Simulating percolating superconductors

Alex Smith

October 29, 2014

Abstract

In this thesis, investigations into the suitability of three 'weak-link' models, designed for the simulation of superconducting cluster systems, are reported. The focus of the investigation is on both the accuracy of the transport properties produced by these models, and the time taken to produce their results.

The thesis develops the theory behind a previous approach which was exclusively used to model percolation systems for coverages p below the critical coverage p_c. The modifications made allow the simulations to extend to system coverages above p_c. An additional 'current-ramping' algorithm, to simulate the systematic increase or decrease of current forced through the system, is described and explored. The results for the three models are compared, and their suitability for future investigations is discussed.

Contents

List of Figures

Chapter 1

Introduction

Nanocluster research has been a burgeoning area of nanotechnology for decades. Transition edge sensors are an intriguing application where the sharp superconducting transition between the superconducing state and the normal state of the sensor makes them an asset for photon detection in certain applications [1]. These sensors can be generated by cluster deposition [2], where clusters are evaporated onto a wafer of silicon. The clusters adhere to the silicon and to each other, forming large networks of randomly placed clusters [3]. This cluster deposition is interesting on its own, with a large number of phenomena arising from the two-dimensional, granular nature of the films produced [4–7]. Of course, deposition is a random process, and so many properties are subject to the stochastic nature of these experiments. The features arising from the random natures of these films can be well described by percolation [8, 9]. However, experimental investigation is a time-consuming and laborious affair. It would be beneficial to prototype and explore the effects of the geometry of the cluster film on the transport properties of the film itself, and how changes in cluster density adjust this, in a quick and simple way. Computer simulation provides a method to investigate the simpler physical features at play in these deposited cluster systems.

In this thesis, a simulation of such a cluster system will be developed from a previous, electron-tunnelling-only model to explore a wider range of cases, specifically those with cluster films able to touch both terminals of the sample, and potential 'weak-link'-based approximations will be compared with a brute force method. The introduction of background information occurs in chapter 2. The original program, forming the basis of the approach used in this thesis, is described in detail in chapter 3. The 'weak-link' models and the current-ramping algorithm are introduced in chapter 4, and the results are presented and discussed in detail in chapter 5.

Chapter 2

Background

2.1 Lattice percolation

Percolation theory became a distinct area of research in its own right in the late 1950s, when Broadbent and Hammersley coined the name and began to establish the mathematics behind the topic [10]. Percolation models are a powerful tool for examining systems that are disordered either in the geometry of the constituent elements, or in the topology and strength of the connections formed [11]. The strength of percolation comes from its focus on system connectivity, and so the interest in percolation arises from the fact that certain properties related to the disorder of particular systems are determined almost entirely by the extent of the system's connectivity. While the properties of percolating systems are often quantified using the concept of coverage p, it can also be measured using the mean path length within the medium $k_F l$ [12], and the differing grain angles of superconducting grains which constitute a more appropriate measure for examining high temperature ceramic superconductors [6, 13].

One of the simplest cases where the influence of coverage on the network can be examined is the lattice. The concept and analysis of lattice percolation is described in detail by Stauffer and Aharony [8], so only the essentials will be addressed here. Take an infinitely large, two-dimensional square lattice, with a light bulb positioned at the center of each square. Randomly select some fraction p of all light-bulbs to turn 'on' and leave the remaining fraction $1-p$ 'off' (Figure 2.1(a)). A coverage of $p = 1$ indicates that all bulbs are in the 'on' state, and conversely, a coverage of $p = 0$ indicates that all bulbs are in the 'off' state. For any p above 0 on an infinite lattice, one expects to find instances of two adjacent squares both being 'on' simultaneously. We consider these adjacent 'on' squares to be connected. As p increases, we not only expect to see more of these connected pairs, but also sets of multiple connected elements. Any set of system elements (in this case, the 'on' squares) which all connect to each other by sharing a common boundary or bond are considered to form a 'Group'. Groups of interconnected sites or bonds are important in the description of transport processes, as transport can only occur between sites if a continuous path exists to connect them together. The properties of 'groups' of networked sites such as the 'strength' of the group - the number of sites within the group - and the

spatial extent of the group can demonstrate important properties of the system as a whole, and so keeping track of which elements of the system are connected to each other is therefore an important task.

An alternative to site percolation is bond percolation, where instead of considering the state of sites, we look at the individual bonds connecting sites together. Again the bonds between sites have a probability p of being on, and a probability $1 - p$ of being off, as shown in figure 2.1(b).

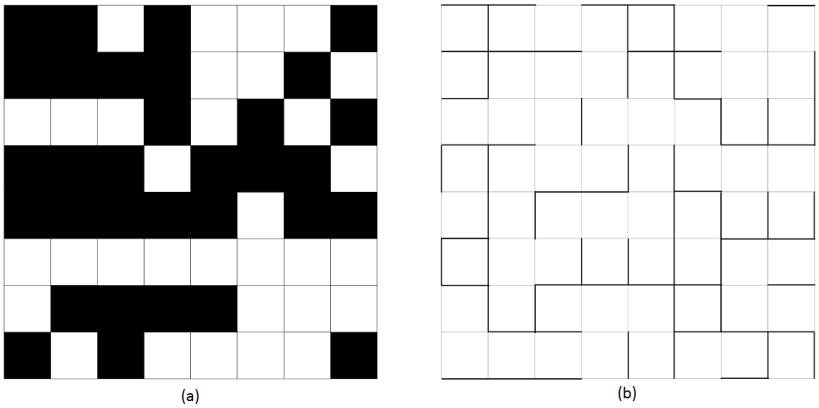


Figure 2.1: (a) Site and (b) bond percolation on a square 2D lattice at $p = 0.5$. Adjacent sites or adjoining bonds are considered to be connected, and form a group.

Site and bond percolation on a square lattice constitute a small sample of the wide variety of available lattice percolation models that exist in 2D, with even more possibilities available in higher dimensions. They still provide a good introduction to some of the key features that should be identified. A good place to start quantifying the connectedness of our sample is by looking for paths that 'percolate' across, or traverse, the system. Once we start looking at the groups constructed using site and bond percolation methods, it is immediately obvious that at some particular coverage (which will be pre-emptively called p_c), the system should contain a series of clusters that manage to form a path that touches two opposing sides of the sample as a single system-spanning group. In a few models, the critical coverage p_c required for the occurrence of the 'phase transition' from disparate groups to a spanning group can be found analytically. For example, in bond percolation on a square lattice (similar to the example shown in Figure 2.1(b)), p_c is found to be equal to 0.5 for an infinite lattice. In other models, such as site percolation on a square lattice, p_c has only been investigated numerically [8]. In finite systems, p_c is defined as the coverage p at which 50% of the constructed systems contain groups which span the network [8].

2.2 Properties of percolating systems

The properties of percolating systems are coverage-dependant, and so are related directly to the coverage p and critical coverage p_c of percolating systems. Two useful concepts relating to the extent of groups, which consist of interconnected elements, are the 'second moment' and the 'correlation length'.

The second moment S of a percolating system is the average number of clusters in a finite group at a particular coverage p, which can be easily calculated. A theoretical value for the second moment is found by first calculating the probability $P_p(s)$ that the group an individual cluster belongs to has s clusters in it, then multiplying it by s. This result is then summed over for all group sizes s in the system. For $p > p_c$, where the largest group spans the system, it is excluded from the sum. $P_p(s)$ is itself found by considering the number of ways a cluster in the system could end up in a group with s clusters, divided by the total number of clusters present in that system.

$$\begin{aligned} P_p(s) &= \frac{s \text{ clusters in group } \times n_s \text{ groups with } s \text{ clusters at coverage } p}{\text{total number of clusters} = \text{sum of all clusters in all groups}} \\ &= \frac{s n_s(p)}{\sum_s s n_s(p)} \end{aligned}$$

With this in mind, the second moment is given by [14]

$$\begin{aligned} S &= \sum_s s P_p(s) \\ &= \frac{\sum_s s^2 n_s(p)}{\sum_s s n_s(p)} \end{aligned}$$

At coverages close to p_c, S follows the power-law relationship

$$S \propto |p - p_c|^{-\gamma} \tag{2.1}$$

where $\gamma = \frac{43}{18}$ [8]. The second moment diverges towards infinity as the coverage p approaches the percolation threshold p_c

The percolation correlation length ξ_p is defined as the average distance between two clusters belonging to the same group [15]. Near the percolation threshold, clusters belonging to the largest groups provide the biggest contribution to the correlation length, and since the number of clusters in the largest group diverges close to p_c, the correlation length also diverges. The correlation length provides a gauge of the homogeneity of the percolating system. The system is reasonably uniform at scales much larger than ξ_p, whereas at length scales smaller than ξ_p, the local geometry is much more important. Near p_c, ξ_p follows the power-law relationship

$$\xi_p \propto |p - p_c|^{-\nu} \tag{2.2}$$

where $\nu = \frac{4}{3}$ in two dimensions [15].

One of the pivotal system properties for the purposes of this thesis is the conductivity of the sample. Take the square site lattice from Figure 2.1(a). If each 'on' square was a square section of a copper sheet, then it would make sense to ask exactly how much current might travel through the sheet if a unit voltage was applied between the left and right edges. Ignoring tunneling currents, one would naturally expect that a copper sheet with no holes ($p = 1$) would display the same conductivity as a whole sheet, and that a missing sheet ($p = 0$) would not be able to conduct any current. From above, the conclusion is that

the sheet will only begin to conduct a current for $p > p_c$, where the sheet has begun to percolate [8]. Experimental [16], theoretical [17] and numerical investigations [18–20] have found that near p_c, the conductivity σ for a metal-insulator composite follows a power-law relationship:

$$\sigma \sim (p - p_c)^t \qquad p > p_c \tag{2.3}$$

The exponent t has been found to be equal to 1.31 in two dimensions to three significant figures [21–24]. When p is close to p_c, a large portion of the squares will not carry much current as they exist far from the optimal path, or lead nowhere - these peripheral bonds are called 'dangling' bonds. Near p_c, the main 'backbone' path carries the vast majority of the current and so dictates the conductivity of the sample [25, 26]. The percolating backbone, and the dangling bonds attached to it, are illustrated in figure 2.2.

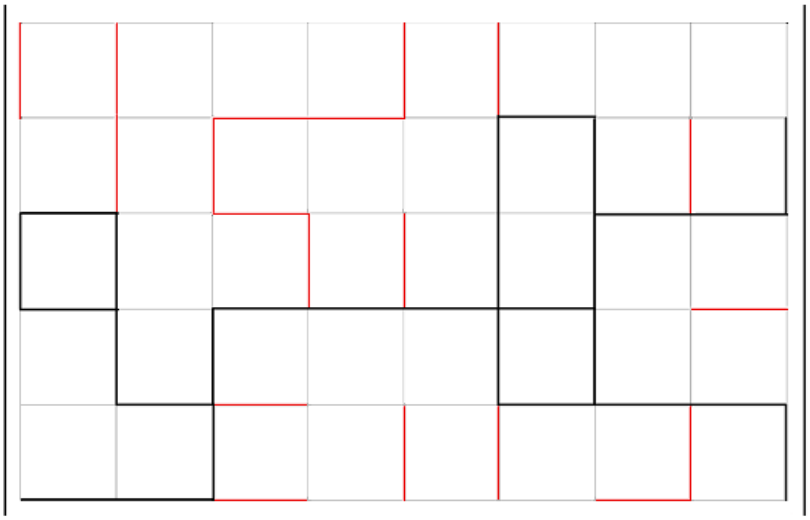


Figure 2.2: A simple illustration of the backbone and dangling bonds of a bond percolation sample. The main backbone of the percolating cluster is coloured black, and will support transport between the left and right-hand terminals. The dangling bonds are coloured red, and cannot support transport.

2.2.1 Nodes, links and blobs

The structure of the network can be approached using the 'node-links-blobs' model illustrated in Figure 2.3 [15], where the nodes are junctions in the network, coupled together by wire-like links and multiply-connected 'blobs' of clusters. The links joining nodes and blobs together are defined as 'red' [27] or 'cutting' bonds, where the removal of the link severs the direct connection between the blobs and nodes that it was previously joining together. The importance of this view comes into play once the differing influences of links and blobs become important, with a good example of this being the investigation of the conductivity of samples. With the 'node-links-blobs' view in mind, one can imagine that the proportions of blobs and links in the '$p \approx p_c$' backbone of bonds would be important. Blobs, being

multiply-connected and thus having a larger number of paths available to carry current, would be expected to have a higher conductivity than links. Additionally, should the current through any individual link be affected by a fuse-like threshold or heating effect, one would expect links to be affected to a larger degree than blobs due to their relative connectivities. In a numerical simulation, finding and mapping exclusively the links near p_c, rather than all links and blobs, can be an advantageous way of quickly predicting the conductivity of the entire sample and simplifying the network as a whole. In any case, in order to calculate the conductivity of the links and blobs, they must first be interpreted as a network of resistors.

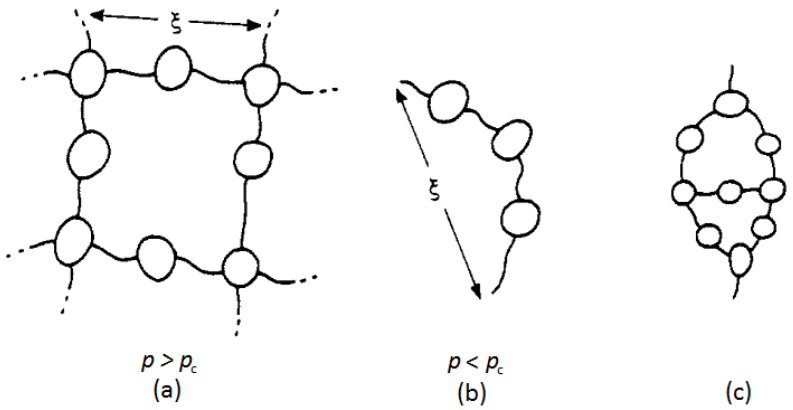


Figure 2.3: The 'nodes-links-blobs' model. (a) Nodes separated by a distance on the order of the percolation correlation length ξ_p are connected by links and blobs. Note that a node may be a blob. (b) Backbone of the incipient infinite cluster below p_c made of links and blobs. (c) Structure of a blob: points separated by a distance b, such that $a \ll b \leq \xi_p$ where a is the lattice spacing, are connected by chains of links and 'blobs' in a self-similar manner. This figure comes directly from the work of Coniglio [15]

2.2.2 Finite-size scaling

Most of the theory surrounding percolating networks bases itself on the assumption of an infinitely large system. As soon as the system size is constrained, it is only possible to use the arguments that were made for the infinite system under certain circumstances. The most significant factor in 2D is that measureable quantities of the whole system act like the infinite system only when the system size L is much larger than the percolation correlation length ξ_p [8] from subsection 2.2. A common approach to obtain accurate infinite-system power-law exponents from finite systems is to apply finite-size scaling [22, 28, 29], a technique that has been applied to a large number of statistical and thermodynamic problems. While most facets of finite-size scaling are well-addressed by the papers collated within reference [30], a short summary of the applicable material will be provided here.

The correlation length of the system ξ_p is known to diverge towards infinity as the system's coverage p approaches p_c [8]. The problem in finite-sized systems,

however, is that the correlation will eventually surpass the size of the system L as the p gets closer and closer to p_c. Given equation 2.2, we can see that for a given coverage p, the system's properties only remain similar to the infinite system while $L \gg \xi_p$, or equivalently, while $L \gg |p - p_c|^{-\nu}$. The point at which the percolative nature of the system breaks down is when the correlation length ξ_p becomes larger than the system size L so that it swallows the system [22].

We can use the idea of the correlation length tending towards the system size to calculate an size-dependant critical coverage $p_c(L)$. In this context, $p_c = p_c(\infty)$. Let us define $p_c(L)$ as the coverage where the correlation length is a similar length to the system size, so that $L \approx \xi_p$. It immediately follows from the equation 2.2 that $L \approx |p_c(L) - p_c|^{-\nu}$, and by extension, that $L^{\frac{-1}{\nu}} \approx |p_c(L) - p_c|$. In other words, the properties of our system start to diverge from those of the infinite system when the coverage p approaches $p_c(L)$, so that $|p_c(L) - p_c|$ approaches $L^{\frac{-1}{\nu}}$.

Alternatively, we can make use the product of $|p - p_c|L^{\frac{1}{\nu}}$. When this product is much greater than 1, the correlation length is much smaller than the system size and the properties of the finite system are similar to those of the infinite system. As p approaches $p_c(L)$, the product approaches 1:

$$|p_c(L) - p_c| \approx L^{\frac{-1}{\nu}} \tag{2.4}$$

$$\text{So then} \qquad |p_c(L) - p_c|L^{\frac{1}{\nu}} \approx (L^{\frac{-1}{\nu}}L^{\frac{1}{\nu}} = 1) \tag{2.5}$$

The point where $|p - p_c|L^{\frac{1}{\nu}}$ approaches 1 is the boundary between the infinite system and the size-affected system. Around 1, the correlation length and system size match, and below 1, the correlation length is larger than the system and appreciable deviations from the infinite system behaviour should be expected due to the inhomogeneity of the sytem. Simulating systems in this regime can still be useful, as it is possible to calculate certain measurable quantities directly from the system size, such as the probability of a randomly selected cluster belonging to the spanning group [8, 31]. Using an empirical scaling model of the form $F(x, y) = y^a f(x/y^{-b})$, several properties of percolating systems can be calculated for infinite systems by extrapolating the results towards $L \to \infty$, such as [8, 22]:

$$\text{Spanning probability } P(p - p_c, L) = f(|p - p_c|L^{1/\nu}) \tag{2.6}$$

$$\text{System conductivity } g(p - p_c, L) = L^{-t/\nu} f(|p - p_c|L^{1/\nu}) \tag{2.7}$$

2.3 Continuum percolation

The constituents of physical networks are usually not constrained to occupying the regularly spaced positions of a lattice. Modifying lattice percolation to respect the variability and random elements of real processes assists in the understanding of a variety of scenarios, including the cluster deposition targeted by this thesis. Instead of giving each cluster a discrete location, the clusters are placed at random in the system, as shown in Figure 2.4. Of course, without having the potential positions and connections predetermined by the lattice in use, the concept of coverage will need to be redefined.

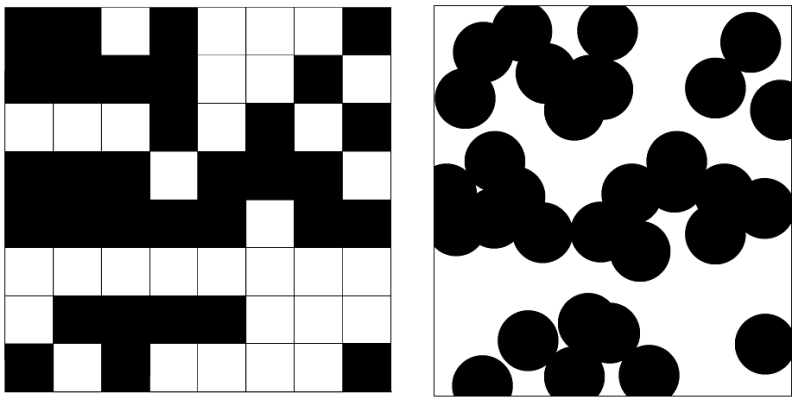


Figure 2.4: (a) The example of site percolation from Fig 2.1 contrasted with (b) a similar continuum system.

As clusters can overlap, we will make use of two types of coverage: the 'first layer coverage' p defined as the proportion of the surface covered by at least one of the clusters that have been deposited onto the surface, and the 'total coverage' p_{total} defined as the sum of the covered areas of all individual clusters. The number of clusters n of area a to be deposited on a surface of area A for a given first layer coverage p, where $a \ll A$, can be calculated as follows:

$$\begin{aligned}
\text{Coverage of one cluster} \qquad & p_1 = \frac{a}{A} \\
\text{Surface fraction not covered by one cluster} \qquad & 1 - p_1 = 1 - \frac{a}{A} \\
\text{Surface fraction not covered by n clusters} \qquad & 1 - p = (1 - \frac{a}{A})^n \\
\text{Coverage of n clusters} \qquad & p = 1 - (1 - \frac{a}{A})^n
\end{aligned} \tag{2.8}$$

The total surface coverage p_{total} of all n clusters with area a, placed on a surface of area A, is given by

$$p_{total} = n\frac{a}{A} \tag{2.9}$$

and so

$$p = 1 - \left(1 - \frac{p_{total}}{n}\right)^n \tag{2.10}$$

Now, as the system size is increased, so that $A \to \infty$, we also have $n \to \infty$ in proportion, so that $\frac{n}{A}$ remains a constant for a particular p_{total}. In this limit, we can see by using the definition of e that

$$\begin{aligned}
p &= \lim_{n\to\infty} \left(1 - \left(1 - \frac{p_{total}}{n}\right)^n\right) \\
&= 1 - \lim_{n\to\infty} \left(1 - \frac{p_{total}}{n}\right)^n \\
&= 1 - e^{-p_{total}}
\end{aligned} \tag{2.11}$$

Thus, by re-arranging,

$$-\ln(1 - p) = p_{total} \tag{2.12}$$

Once the number of clusters required for a particular first layer coverage has been calculated, they can be placed on random positions on the surface that are generated using a Poisson process [32] so that the expected density of clusters in a randomly chosen region of area A is the same for any region with the same area, regardless of the area location. A method that achieves this [33] is detailed in section 3.1. The connections and conductivity can then be determined by investigating the network formed by the touching clusters and assigning conductivities to each of these connections. The continuum percolation model has a different value of p_c when compared with those of the various lattice models, approximately equal to 0.676339 [34].

Fortunately, the continuum model still shares some properties with the lattice model, particularly in two dimensions. In order to uncover the conducting paths within a continuum network, and thus calculate the conductivity, one can construct a network roughly equivalent to a lattice network by performing a Delaunay Triangulation, using the cluster centers as points [35]. For a conducting sheet with holes randomly punched into it [16], one can instead construct a Voronoi Tesselation using the set of hole centers [23, 36]. This remapping serves the additional purpose of allowing us to simplify the complicated structure of disordered clusters in the continuum and approximate it with an appropriately constructed lattice. Instead of attempting to deal with the complexities of the potential electronic structure of the system, we can treat the bonds from the simplified structure as resistors in order to calculate the conductivity of the continuum network. The triangulated network of resistors produced by the Delaunay triangulation remapping demonstrates the same conductivity relationship as the lattice case, described by Equation 2.3. More importantly, in 2D they share the same t value of 1.31 [37]. This new network is a 'random resistor network', which will be discussed in subsection 2.5.1.

Continuum percolation remains an area of active research. While the conductive properties of continuum systems have been investigated by lattice-based approaches for some time, as will be discussed in section 2.5, the investigation of other facets of the continuum requires a fully continuum simulation rather than a lattice approximation. Recent research into continuum systems include numerical investigations of the percolation threshold and power-law exponents of objects with different shapes [38–41], the properties of continuum systems containing disks with a power-law distribution of radii [42], the fragmentation of percolating groups by removing critical bonds [43] and the application of tunnelling currents to continuum systems [4, 44].

2.4 Superconductivity

Certain materials present unusual conductivity characteristics when their temperature T is reduced below some material-specific critical temperature T_c. In the vast majority of cases, the unusual characteristic in question is the sudden disappearance of the material's electrical resistance. This phenomenon is called 'superconductivity'. While the theory is covered in reasonable detail in reference [45], a short summary of the relevant details will be presented here. There

are two significant features of a bulk superconducting material. The first is perfect conductivity. With T below T_c, the resistance of electronic transport within the material disappears. The second feature is perfect diamagnetism. Not only is magnetic flux stopped from penetrating the superconductor, which would be another demonstration of perfect conductivity, but magnetic flux is also ejected from the interior of the superconductor. This ejection of magnetic flux is known as the Meissner effect. The ability to exclude magnetic fields can be overcome by a suitably large magnetic field H, which will extinguish the superconducting state of the material if the incident field strength surpasses some critical value H_c. In a similar way, as electric current generates its own magnetic field, a superconductor can destroy its own superconductivity if it carries a current density J in excess of its critical current density J_c. While temperature and incident magnetic flux are certainly significant in affecting the function of superconductors, they are outside the scope of this thesis, and will not be directly treated here.

Conventional superconductivity, as originally investigated by Kamerlingh Onnes [46, 47], is well described at a microscopic level by Bardeen-Cooper-Schreiffer (BCS) Theory [48]. In BCS theory, superconductivity is the result of electrons being bound together to form 'Cooper pairs' when some attractive force between them exceeds the repulsive Coulomb force. For a conventional superconductor such as lead, the attractive force in question is produced by a phonon-mediated polarisation of the underlying medium. For example, an electron in the conduction band of a metallic substance might attract the surrounding positive metal ions. The excess of positive ions attracts the second electron, drawing the two electrons together [49]. The effective size of a Cooper pair is given by the temperature-dependant superconducting coherence length $\xi_s(T)$, a length that is generally many times larger than the lattice spacing of the underlying metallic lattice.

Ginzburg-Landau (GL) theory [50] is a macroscopic theory of superconductivity that describes the Cooper pairs using a pseudowavefunction $\psi(\mathbf{r}) = |\psi|e^{i\phi(\mathbf{r})}$, called the 'complex order parameter' , which can be used to find the number density n_{sc} of the superconducting pairs [51]. The importance of the phase ϕ of the superconducting wavefunction comes forward when one begins to investigate the nature of resistance in one/two dimensional superconductors. GL theory describes the supercurrent $\mathbf{J}$ using the formula [45]

$$\mathbf{J} = \frac{e^*}{m^*}|\psi|^2(\hbar\Delta\phi - \frac{e^*}{c}\mathbf{A}) \tag{2.13}$$

where e^* and m^* are the effective charge and mass of a Cooper pair, ψ is the complex order parameter, $\Delta\phi$ is the phase difference and $\mathbf{A}$ is the applied electromagnetic vector potential.

2.4.1 Josephson junctions

The Josephson effect describes the ability of supercurrent to travel through 'weak links' separating two superconducting elements [52]. The weak link in question can be any reasonably thin barrier that is likely to provide resistance. Insulating

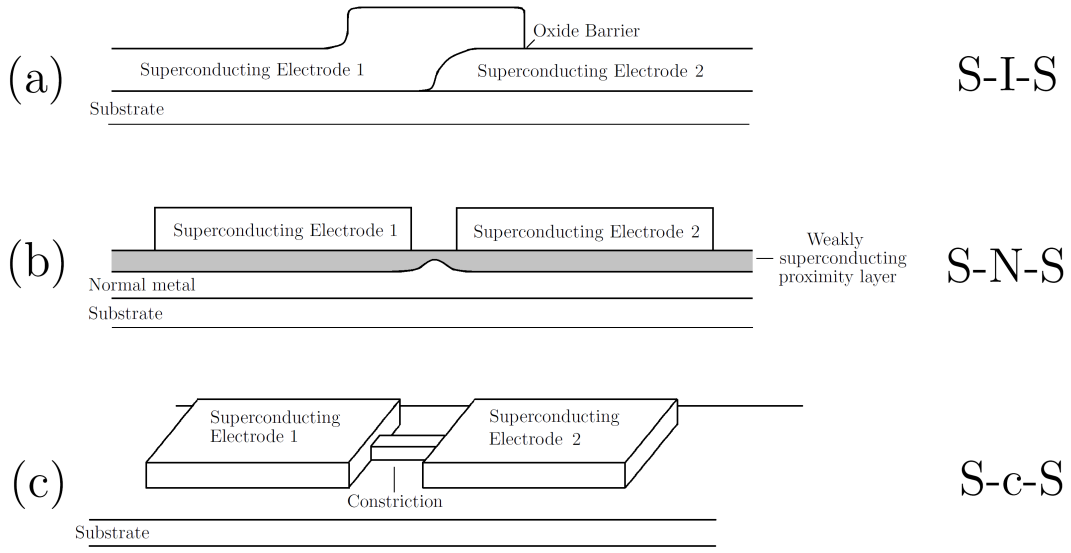


Figure 2.5: Three types of Josephson junctions [45]. (a) represents a superconductor-insulator-superconductor (S-I-S) Josephson junction, with a metal oxide barrier acting as an insulator between the two superconductors. (b) represents a superconductor-normal-superconductor (S-N-S) Josephson junction with the normal metal element, turned weakly superconducting due to its proximity to the two superconductors, acting as a poor conduit between the two superconductors. (c) represents a superconductor-constriction-superconductor (S-c-S) Josephson junction with the narrow constriction of superconducting material slowing the flow of supercurrent between the much thicker superconducting elements to either side

oxide layers, metals rendered weakly superconducting by a proximity effect or narrow constrictions of the superconducting material all provide suitable scenarios where the Josephson effect might present itself. In all three cases, the optimal distance between the contacts d should be less than the superconducting coherence length ξ_s [53]. Constrictions should be one-dimensional, and so should have cross-sectional dimensions that are also smaller than ξ_s. In high-temperature superconductors, the boundary between neighbouring grains is enough of a barrier to bring the Josephson effect into action. A device designed in this manner specifically to produce the Josephson effect is a Josephson junction. Three junction designs are presented in Figure 2.5.

Two equations govern the Josephson effect. A Josephson junction can maintain a zero voltage supercurrent I_s that depends on the phase difference $\Delta\phi$ across it:

$$I_s = I_c \sin \Delta\phi \tag{2.14}$$

In this context, I_c is the biggest supercurrent that the Josephson junction can support without going normal.

The change in phase difference with respect to time is governed by the voltage across the junction:

$$V = \frac{\hbar}{2e} \frac{d(\Delta\phi)}{dt} \tag{2.15}$$

Josephson junctions can be classified depending on their capacitance. Of interest to this thesis is the case of overdamped Josephson junctions, which have a low capacitance between the two superconducting elements forming the junction. The cluster interactions responsible for electrical transport in the experimental scenario feature low capacitance, making overdamped Josephson junctions a suitable candidate for modelling. The current-voltage characteristics associated with overdamped junctions can be described using [54]

$$V = R(I^2 - I_c^2)^{\frac{1}{2}} \tag{2.16}$$

Underdamped junctions have been avoided as they are hysteretic and so would obfuscate potential hysteresis due to the percolating network, making them an unwieldy device for diagnosing system properties during simulations.

Finally, masses of Josephson junctions can be constructed in lattice arrangements by lithographic processes [55, 56] to form arrays. Josephson junction arrays have been reasonably well investigated in the literature [57–59], and as junction arrays can be approached from within percolation models, they are applicable to this thesis. As percolating cluster systems involve connections that are on the same scale as the superconducting coherence length, connections between clusters might be expected to act like Josephson junctions [60, 61].

Two important power-law relationships relate specifically to percolating superconductor networks. Firstly, in a paper by Octavio et al. [62], two possibilities are presented for the critical currents of individual S-c-S Josephson junctions located in a percolating array. If the width w of the constriction exceeds the superconducting coherence length ($w > \xi_s(T)$), then the critical current is governed by a depairing mechanism where $I_c \propto w$. Alternatively, if $w < \xi_s(T)$, then the neck width is Josephson-like, and the critical current is a constant, regardless of neck width. This is important factor in the evolution of the system's critical current as the coverage increases. Octavio et al. discuss that the critical current density J_c of the percolating system should vary with the coverage p as a power-law, governed by an exponent v:

$$J_c \propto (p - p_c)^v \tag{2.17}$$

Octavio et al. simulated the two possibilities of depairing and Josephson-based critical currents on a square lattice of superconducting/conducting bonds, and found that the difference in local critical current resulted in a different exponent v for the two cases. While the Josephson-based critical current produced an exponent of $v = \frac{4}{3}$ equal to t from the conductivity problem, the depairing critical current produced an exponent of $v = 1.821 \pm 0.01$, almost 0.5 greater than the Josephson case. This value is corroborated by Lee et al. [63], who suggest an exponent of $v = 1.7$.

Secondly, in percolating systems, the voltage V measured across the system is related to the current density J and critical current density J_c by [60]

$$V \propto (J - J_c)^a \tag{2.18}$$

The value of a is determined to be equal to 2 by Lee et al. [63], and equal to 2.1 by Granato et al. [60].

2.4.2 Low-dimensional superconductors

When the sample of superconducting material is constrained to a one-dimensional wire or a two-dimensional plane, the mechanisms for resistance generation within the sample are discussed in terms of phase in the literature. As seen in the case of Josephson junctions, the GL theory phase difference $\Delta\phi$ between adjacent superconducting elements is an integral part of describing the supercurrent flow between superconducting regions. A reduction in the phase difference can lead a corresponding drop in the size of the supercurrent, and so to the appearance of resistance between the two elements.

This is significant in one and two-dimensional superconducting systems, where the dimensionality is found by comparing the size of the superconducting system along its various spatial dimensions (x, y and z) to the superconducting coherence length ξ_s. A system is considered to be one-dimensional if two of its spatial dimensions x and $y < \xi_s$, with $z > \xi_s$. A superconductor is two-dimensional if $x < \xi_s$, and y and $z > \xi_s$. In a one-dimensional wire, the concept of phase slips becomes important to the description of the system's resistance, as random fluctuations can cause a spontaneous 2π change in the phase of the complex order parameter [64], leading to a change in the supercurrent. In a constant voltage system, phase slips can occur randomly, reducing the phase gradient by 2π and thus the supercurrent by an amount dictated by equation 2.13.

In two dimensional systems, it is possible for superconducting elements, arranged in a circle around a defect such as a hole and with a magnetic field incident on the defect, to have the phase gradients between adjacent elements arranged so that the total phase difference around the loop is a multiple of 2π. In this scenario, the supercurrent travels around the loop in a vortex. At low temperatures near 0K, below the 'Berezinskii-Kosterlitz-Thouless' (BKT) [65,66] transition temperature T_{BKT}, vortices of opposing helicity exist in bound pairs, held together by the attractive potential between them. The BKT transition describes the spontaneous splitting-up, or 'depairing' of vortices at temperatures above the threshold $T_{BKT} < T_c$ [45] due to the associated increase in entropy making their dissociation more energetically favourable at higher temperatures. With an applied current, these vortices can be driven across the system, generating resistance due to 'flux flow' in the case of continuous films, or 'flux creep' in systems with distributed defects or pinning sites [67].

Disordered systems, such as the percolating lattice described in subsection 2.1, exhibit properties of both one-dimensional wires and two-dimensional planes with randomly distributed defects, and so present the possibility that both vortices and phase-slips influence the resistive properties of the system. Additionally, if the sample is allowed both normal and superconducting domains, then the ability of supercurrent to tunnel between superconducting elements is another important element of the physical scenario and so the Josephson effect must also be considered. While a truly accurate simulation of the physical scenario should include the potential for all three of these phenomena to occur, this thesis will restrict itself to the Josephson effect.

2.5 Calculating the conductivity

2.5.1 Random networks

A simple way to retrieve the essential conductive properties of these networks from a theoretical perspective is to treat every connection in the system, whether it is a link or a part of a blob, as a resistor. The network produced is a 'Random Resistor Network' (RRN), and can be examined either theoretically or numerically. The RRN model is a useful way to examine the exponent relationships of percolating systems via numerical investigations [19], as Kirchoff's laws can be readily calculated for a network of resistors with an applied voltage to produce the conductivity of the sample. The RRN model allows the investigation of the conductivity-coverage relationship described in equation 2.3, and can be applied to model a sheet of a metal-insulator mixture in this manner, with the metallic portion covering p of the sheet and the insulating portion covering $1-p$ of the sheet. This scenario can be replicated on a bond-percolation-style grid with metallic conductors of conductance σ along a fraction p of the bonds and insulators with conductance 0 along the remaining fraction $1-p$. This new RRN can be investigated by applying Kirchoff's laws and solving the remaining system with linear algebra. When p approaches p_c from above, the conductivity σ of the system approaches 0 [68]. As p increases, the number of conducting links in the network increases. The conductivity of the system depends on both the topology of the system itself and the conductances allowed for each individual link. RRNs have been approached using renormalisation [26, 69] the Effective-Medium approximation [70, 71] and Mean Field theory [72]. The conductance of each link can be made identical, allowed to vary due to various perturbations [73] or allowed a distribution of conductances [23].

An important alternative mixture is a superconductor-metal mixture. With the advent of high-temperature ceramic superconductors, the application of percolation theory to metallic wires with embedded superconducting elements, or superconducting grains [6, 74], has stemmed in part from the earlier investigation and modelling of superconductor-metal mixtures [68]. One can model these systems by randomly setting the bonds of a lattice in the same way as the random resistor networks, except with an increasing coverage p of superconducting links with an infinite conductance and a decreasing proportion $1-p$ of metallic resistive links with a finite conductance. The conductivity of the system σ diverges as the fraction of superconducting elements p approaches p_c from below. The random network produced using superconducting and metallic connections is called a 'Random Superconductor Network' (RSN). The conductivity of the superconducting network is also a power-law relationship [63, 75]:

$$\sigma \sim (p_c - p)^{-s} \qquad p < p_c \tag{2.19}$$

In two dimensions, the exponent s is equal to t (where $t = 1.3$) [28].

More complicated features can arise from expanding the resistor types from the usual dichotomy to three or four-part mixtures involving insulator, conductor, superconductor and tunnelling behaviour, or by introducing additional facets of

superconductivity theory, such as Josephson junctions and the various phase transitions (such as Kosterlitz-Thouless transitions) that arise from their presence. Multi-component percolation has been investigated on lattices under the guise of polychromatic/AB percolation [76–79], where only sites of opposing 'colours' are linked together to establish transport bonds [80], or with the multiple components being given seperate conductivities and individual coverages [81–83].Josephson-based percolating systems have been investigated in the context of both disordered Josephson arrays [60, 61, 84–88] and granular superconductors [5–7, 89–92]. The dynamics of systems based on the Josephson equations are determined by the phase differences $\Delta\phi$ across Josephson junctions or between superconducting grains. In each of these areas, the focus has been decidedly on the dynamics of the lattice, and treatments of continuum systems directly have been avoided. While disordered Josephson junction arrays are capable of producing the physics associated with superconductivity, they ignore the influence of cluster geometry on the resistances and critical currents of the connections.

2.5.2 Calculating conductivity in lattice models

Once the geometrical construction of percolating networks is achieved and the appropriate RRN or RSN constructed, the task of simulating the electrical transport properties of these systems is the next step. The methods required to calculate the conductivity of a random network are reasonably simple for the lattice, and can be extended to operate on the continuum models.

In order to determine conductivity of a random network, we first need to give a conductance g to each connection between neighbouring nodes. The potential values for the conductance g of the network connections are selected with a certain probability from a chosen distribution of conductances. The distribution can be chosen to cover a wide range of values, or simply be a dichotomy between two constant values. The simplest approach for calculating the conductivity of a random system is to assume that it is arranged in a lattice, as discussed in section 2.1. The two important cases to consider for lattice systems are RRN, where $g = 0$ or $g = c$ for a constant conductance value c, and RSN, where $g = c$ or $g = \infty$. In each of these cases, we have only two possible options for the conductivity of a particular connection. We can therefore investigate the effects of placing different proportions of the two components at random positions in our system by adjusting the coverages of the components in our percolating network. The replacement of connections with resistors is illustrated in Figure 2.6. Conventionally, the proportion of the system occupied by the higher-conductance bonds is given by the coverage p. Conversely, the proportion of the system occupied by the lower-conductance bonds is given by $(1 - p)$.

There are several approaches that can be used for calculating the conductivity in a lattice network. Electrical conductivity is an example of a transport process, similar to the conduction of heat or molecular diffusion. As such, calculating the conductivity has been treated as a diffusion problem in the past [93–95]. However, the classic approach is to investigate the conductivity produced as a result of solving Kirchoffs equations [8]. The focus of this section, and the remainder of

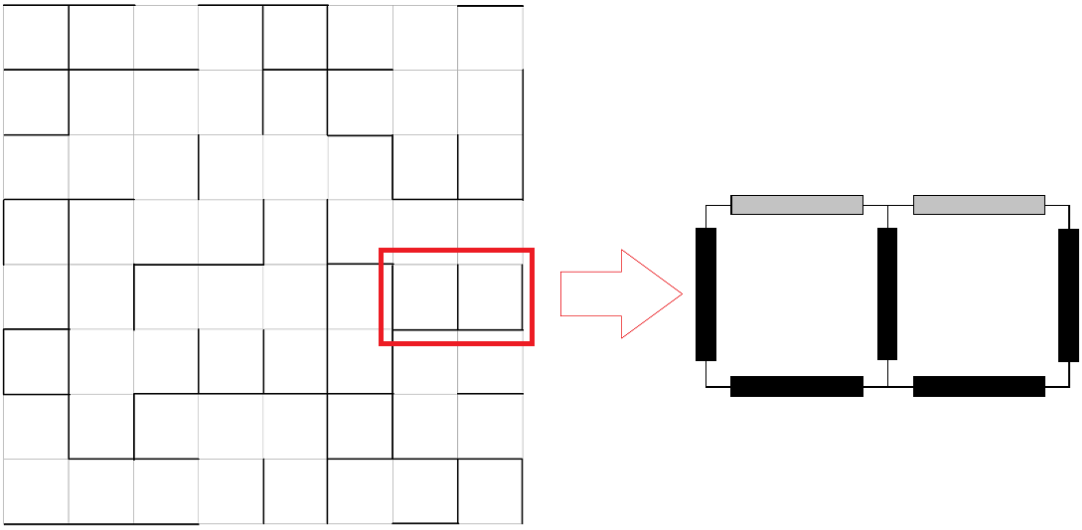


Figure 2.6: An illustration of assigning conductances to the connections between nodes, where each bond is replaced with a resistor. On the right, grey resistors represent connections with low conductivity, while the black resistors respresent connections with high conductivity.

this thesis, will be on the Kirchoff solution papers.

For a percolating lattice of sites or bonds, the process of finding the conductivity in this manner is a simple procedure. In bond percolation, one constructs a series of simultaneous equations detailing the current flowing into and out of each node through each of the bonds connecting it to its neighbours. In site percolation, the same can be done for sites by looking at the site boundaries. A key assumption is that each node or site is divergence free, so that all of the current flowing into a node also exits the node, and vice versa. In the same way, we also assume that the entire system is divergence free, so that the amount of current flowing into the system through one terminal is also the amount that leaves the system through the other terminal.

Ohms Law states that $I = Vg$ for a resistor, where I is the current through the resistor, V is the voltage across the resistor, and g is the conductance of the resistor. This can be used to calculate and state the total current I into and out of node i from each of its neighbours using the voltage difference between adjacent nodes and the conductance between them, as represented in equation 2.20 by the sum through j. We must also take into account the current supplied to, or drawn from, node i by the terminal ($I_{terminal}$), which will be equal to 0 for any node that doesn't touch the terminal. Thus, the divergence of node i can be quantified as follows:

$$I_{terminal} + \sum_{j} g_{ij}(V_j - V_i) = 0 \tag{2.20}$$

The basis for equation 2.20 is illustrated in figure 2.7. Two different nodes have been selected to demonstrate the divergence condition, one in the middle of the system and one along the terminal. The node in the center of (a) is positioned in

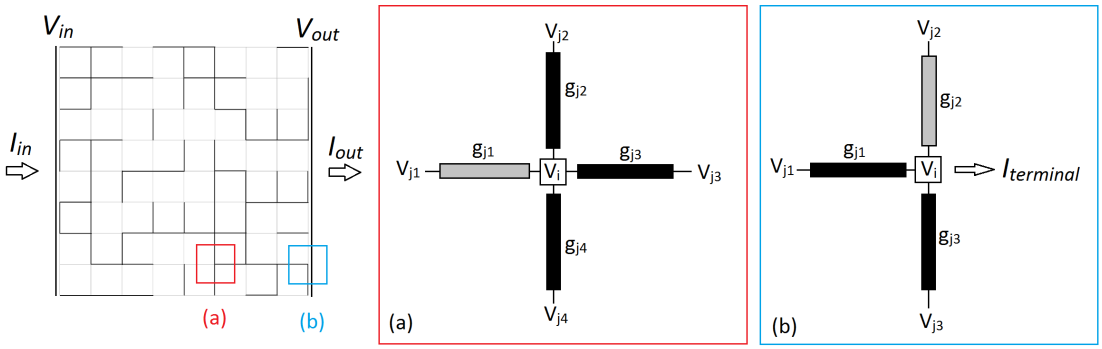


Figure 2.7: Calculating the conductivity of the random network. The nodes in the network i are required to have the amount of current entering them be equal to the amount leaving. This is regardless of whether the node is located in the center of the network (a) or along a terminal (b)

the middle of the system, so there is no connection to the terminal and thus no $I_{terminal}$ component in the equation. It is bordered on all sides by nodes, in this case labelled $j1$, $j2$, $j3$ and $j4$, and so the sum through j must account for all four of these connections. g_{ij} indicates the conductivity of the bond connecting the central node i to neighbouring node j. V_i indicates the voltage difference between node i and 0.

In order to obtain a solution, boundary conditions must be set at the terminals. We can detail the currents $I_{terminal}$ distributed to each terminal node by the terminal. The sum of these currents is the I_{in} term in Figure 2.7. The alternative is to require that there be a particular voltage drop between the two terminals, so that V_i for any group i along the input terminal is some constant input voltage V_{in}, and the V_i for any group i along the output terminal is V_{out}. In either case, for the result to be sensible, it is necessary to ensure that the same voltages are assigned to every node along both the input and output terminals respectively. Knowing the arrangement of the conductances g_{ij} between all nodes in the system, enough information is available to find the voltage at every node and the current between each node and its neighbours by linear algebra, which will be discussed in section 3.3. One can then find I_{in} or I_{out} for one of the two terminals and reapply Ohms law to find the conductivity between the two terminals:

$$\frac{I_{in}}{V_{in} - V_{out}} = g_{system} \tag{2.21}$$

Computing exact solutions to Kirchoffs equations can be difficult and time-consuming for large systems. It is possible to relieve some of the computing power and memory requirements by using relaxation methods, where an initial guess at the solution is made and iteratively adjusted to improve the accuracy [96]. The main alternatives that appear in the literature are the Transfer-Matrix and large-cell renormalization group approaches. Both of these approaches attempt to reduce the number of nodes that need to be processed at once by dealing with the system in small blocks first, then combining these blocks into larger blocks. Regardless of approach, this can be expedited by removing the dangling bonds [15]. These dangling bonds consist of nodes and connections that will not assist with conducting current because they belong to a group of nodes that has only a single

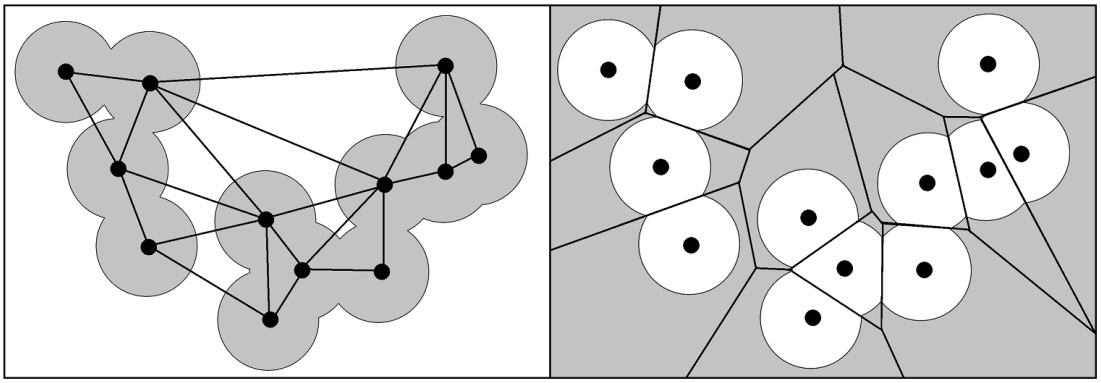


Figure 2.8: The Delaunay triangulation (left) and Voronoi tesselation (right) of a set of points.

bond connecting it to the percolating backbone. As all current can only enter or exit the group through the same bond, the net current flow through the bond must be zero, and so the dangling group will not conduct any current. In the case of the transfer matrix approach [96–99], the transfer matrix is constructed at the same time as the lattice is constructed, and is altered as the conductance of each bond is chosen and put in place. The current between particular sections of the lattice can then be calculating using appropriate entries in the transfer matrix as an application of standard transfer matrix theory. In the large-cell renormalisation group approach [24, 100, 101], the method involves introducing a ghost cell that couples exclusively to the connections that make up the percolating backbone. The process investigates the ability of current to make its way to the ghost cell, and use this information to collapse square sections of bonds down into a single bond with the net conductance of those that it consumed. Suffice to say, references [24, 96–101] represent the bulk of the attempts to calculate the conductivity exponent t, and have lead to reasonably accurate measures of the conductivity of percolating systems in the past.

2.5.3 Calculating conductivity in continuum models

Even when allowing the nodes in the network to take random positions, as is the case in continuum percolation, the previous lattice approaches remain useful, because the continuum percolation conductivities can be calculated by mapping the conducting connections onto a lattice with appropriate bond strengths. In the case of the Swiss-cheese model where the discs are insulating, the lattice bonds in question are found by constructing a Voronoi tessellation on the disc centres. In the inverse-Swiss-cheese model, the lattice is instead made by constructing a Delaunay triangulation involving the disk centres.

An example of the networks produced by the Delaunay triangulation and the Voronoi tesselation of a set of discs is shown in figure 2.8. The Voronoi tesselation divides the system into polygons so that each polygon only contains one disc center. The edges of each polygon surround all points in the system that lie closer to the disc center belonging to that polygon than any other disc center [102, 103]. The Delaunay triangulation, on the other hand, joins three disc centers together

by edges if, and only if, a circle constructed to pass through all three disc centers does not contain any additional disc centers [35, 104, 105].

In the Swiss cheese model, any current flowing through the system must follow a path that avoids the discs. It can be shown that the system conductivity can be calculated as a bond percolation problem involving the edges of the Voronoi tesselation of the discs [23, 102]. In the inverse Swiss cheese model, where the current must travel through discs, it can be shown by a similar argument that the system conductivity can be calculated using the Delaunay triangulation of the system [23,35,103]. Given the focus on cluster depositions in this paper, with the deposited clusters being modelled by randomly placed discs, the Delaunay approach is the most appropriate for our purposes and so will be the remaining focus of this section.

The conductance of each of the bonds is determined by the thinnest cross-section in the continuous conducting material along the bond line. If the bond is forced to cross through a region of insulating material then the assigned conductivity is 0 [23,37]. For the Delaunay triangulation, this is the same as saying that the bond conductivity $g = 0$ if the bond length d is greater than the sum of the disc radii $r_1 + r_2$ for the two points in question. Once the Delaunay network is constructed and bond conductivities have been assigned, this newly-created lattice system can have its conductivity calculated using a Kirchoff solution approach identical to the approach utilised in the cubic lattice case.

In reference [23], Feng et al. use the remapping approach discussed above to determine that the structure of the continuum system determines the probability distribution of various bond strengths in the lattice mapping for the inverse Swiss cheese model. Their argument continues that the conductivity of a particular continuum model can be investigated by employing a lattice with bond strengths randomly assigned using the distribution produced by that continuum model. Additionally, with reference to [36], they note that the geometric connectivity of the Swiss cheese and inverse Swiss cheese models is unchanged from the discrete lattice models, and so the systems only differ by the distribution of bond strengths given to the connections.With this in mind, the techniques used to calculate the conductivity of normal lattice systems (particularly to take advantage of inadequate computing power) can be easily reapplied to the new case.

The fact that it is such a simple job to choose a distribution of bond strengths and assign those to the bonds of a lattice to replicate continuum systems has led to investigation into the effects of both conventional distributions of bond strengths, such as the distribution associated with the inverse Swiss-cheese model, as well as with more unusual distributions. Obviously, if the distribution of bonds on the square lattice is two-valued so that bonds are assigned conductivities of either c or 0, then the random resistor network case is replicated and so the conductivity exponent t remains equal to 1.3. The question exists: for which distributions does this remain true? And for which distributions does the conductivity exponent lose its universal value? The argument [23] is that, for certain continuum models such as the inverse Swiss cheese model, the probability of selecting a high bond strength is dictated by an exponential relationship, so that the probability of picking a particular bond strength $P(g)$ looks like $P(g) \propto g^{-\alpha}$ where the value of α is chosen

to fit the particular system. From this, the values of α appropriate for different continuum systems can be investigated and compared with the continuum case. For the inverse Swiss cheese model in two dimensions, it has been shown [23] that the conductance of each bond should be equal to a constant value c, independent of the neck width of the overlapping discs. This is the equivalent of setting $\alpha = \infty$. Thus, there is no difference in the conductivity characteristics between the square lattice and disc model in 2D and so the value of the conductivity exponent t should retain its universal value of 1.3.

Generally speaking, any continuum distribution with a distribution specified by $\alpha < 0$ exhibits the universal conductivity exponent [106]. For more unusual percolation models, where the probability $P(g)$ of a bond having a conductance of g is given by $P(g) = g^{-\alpha}$ with $\alpha > 0$, nonuniversality of these exponents becomes an important part of the model, and the value of t should be adjusted to $t = t_{un} + \frac{\alpha}{1-\alpha}$ [106–108]. Simulations of percolating systems involving other shapes such as sticks, squares, rectangles and ellipses have been performed to investigate this. The additional degrees of freedom in three dimensional systems provide a sufficient mechanism to lose the universality of t [109, 110]. In 3D, simulations have been performed using mapping techniques similar to the 2D case. For the more unusual systems such as percolating ellipsoids, the space is discretised beforehand [111]. A comparison has been made between mapping approaches and a truly continuum field calculation [112], but an extension to inverse Swiss cheese systems has yet to be published.

Chapter 3

Simulating percolation and system conductance

This chapter will detail the original program that formed the basis of this thesis' work. The original program, developed by James Carr and Shawn Fostner, was previously used to simulate the deposition of clusters and the conductivity calculation for systems where there was no spanning group, so the conduction of electricity was treated as a tunnelling problem. Their approach is limited to systems where $p < p_c$, and is discussed here, progressing through four main steps:

1. Cluster deposition
2. Cluster grouping
3. Conductance assignment to connections
4. Solving for the system conductance

3.1 Deposition

The cluster deposition constructs a deposited cluster sample for a desired input coverage p and a selected size L of the system, measured in terms of cluster diameters. For example, a system size of 200x200 indicates a sidelength of $L = 200$ cluster diameters. Given a distribution of cluster radii, the average coverage area of a single cluster is introduced into equation 2.12 to work out the 'total cumulative coverage' p_{tot} and the number of clusters required in the deposition to approximately produce the desired first layer coverage p.

The clusters are deposited using a Poisson point process as described in reference [32]. In short, a Poisson point process constructs a system of randomly placed points with a uniform density by splitting the system up into subdomains, allocating the number of points to go into each subdomain using a Poisson distribution, then providing each point with a location randomly chosen from a uniform distribution of points found in the subdomain.

Using this process, the system is divided into an array of squares, each with

side lengths of the biggest available cluster diameter $2r_{max}$. In this thesis, all clusters will be given the same diameter of $2r = 1$.Next, the number of clusters expected in each square of the subdivided system is determined using a Poisson distribution. The probability $\text{Pois}(x)$ of finding x clusters in a particular square is

$$\text{Pois}(x) = \frac{\langle x \rangle^x e^{-\langle x \rangle}}{x!} \tag{3.1}$$

where the average number of clusters $\langle x \rangle$ in each square is determined by the total number of clusters in the system n, divided by the number of squares in the array. For example, in a 200x200 system about to receive 60,000 clusters, each 1x1 square in the system would receive 1.5 clusters on average, and so $\langle x \rangle = 1.5$. In order to choose how many particles go in each square subdivision, a random number generator is used to randomly select a number n_r from the interval (0,1]. The set of values of n_r that correspond to the choice of selecting x clusters for the subdivision is given by $P(x)$, which we will define as follows:

$$P(x) = \begin{cases} (0, \text{Pois}(0)] & \text{if } x = 0 \\ (P(x-1), P(x-1) + \text{Pois}(x)] & \text{if } x > 0 \end{cases} \tag{3.2}$$

Thus, if the random number n_r that is produced is found to lie in the interval $P(x = a)$, then a clusters will be placed in the chosen subdivision.

Each of the cluster introduced to the chosen subdivision is given a randomly selected x-y position in the square, and a randomly-selected radius taken from the radius distribution provided. In this thesis, the radius provided is always $r = \frac{1}{2}$. An illustration of a single subdivision following deposition is shown in figure 3.1(a). The position given is stored as a complex number in a vector of positions, where the (x, y) coordinates given to each cluster are stored as $z = x + iy$. This is both to save on memory usage and access time, but also to allow for simpler distance calculations. x and y can still be accessed individually as $x = Re(z)$ and $y = Im(z)$.This process is almost identical to the approach used in the literature [33, 113]. The radius of each cluster is also stored in a vector. Every property of the cluster is saved using the order in which the clusters are placed. For example, the m^{th} particle that is placed into the system will be associated with the m^{th} entry in the position vector, the m^{th} entry in the radius vector, and so on. The position m of a particular cluster's properties in the various vectors describing the system will be referred to as the cluster's 'label'.

3.2 Grouping

Given the locations and radii of the clusters, the next step towards finding the conductivity of the system is to work out which clusters are connected to each other in a group. By identifying the unique interconnected group that each cluster belongs to, it is possible to quickly uncover whether current can be transferred between any two clusters, as the transfer of current is only possible when the clusters are directly touching (in the absence of tunnelling). Discovering which clusters are connected together can be done by going through each subdivision

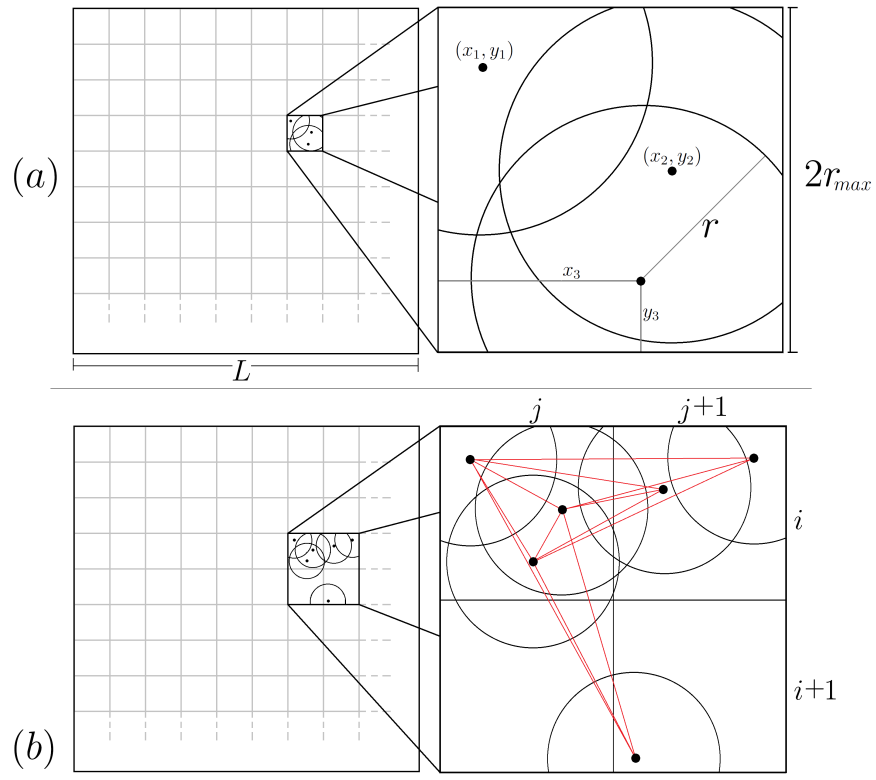


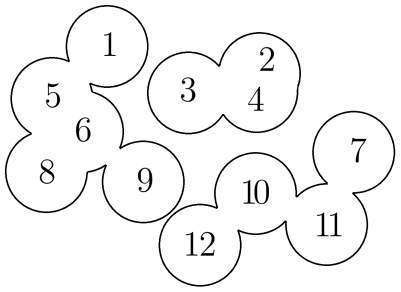
Figure 3.1: (a) demonstrates the deposition of clusters in a particular square subdivision (i, j). The number of clusters in the square is randomly chosen using equation 3.1 - in this case, there are 3 clusters deposited. Each cluster k is assigned a random location (x_k, y_k) within the square. In this case, each cluster is given the same radius of r. (b) demonstrates the process of discovering which clusters touch one another. The clusters in subdivision (i, j) are compared with each other and with their neighbours in subdivisions $(i, j + 1)$, $(i + 1, j)$ and $(i + 1, j + 1)$. Clusters touch if the distance d between them, represented here by the length of the red line connecting them, satisfies $d \leq 2r$.

in turn and looking at the connections made between the clusters in the chosen subdivision and its neighbouring subdivisions.

Clusters that are further apart than the biggest cluster diameter of the system, $2r_{max}$, cannot possibly touch each other. Due to the fact that the square subdivisions have been given a sidelength of $2r_{max}$, every cluster that can possibly come into contact with a chosen subdivision (i, j) must be located in the eight squares surrounding (i, j). In fact, as long as we plan to look at every subdivision in the grid, we only need to look at three of the eight neighbouring squares - $(i, j + 1)$, $(i + 1, j)$ and $(i + 1, j + 1)$ - in addition to the square (i, j) itself in order to find every connection [33].

To find clusters within touching distance of each other, we select each square (i, j) in turn. For square (i, j), we consider every combination of cluster pairs involving the clusters in (i, j), by pairing every (i, j) cluster with each of the clusters in subdivisions (i, j), $(i, j + 1)$, $(i + 1, j)$ and $(i + 1, j + 1)$. This process is illustrated in figure 3.1(b), where each pair is depicted as a red line joining the two paired clusters together. We then compare each of the distances between

paired clusters with the sum of the cluster radii. If the distance between the two clusters is greater than the sum of their radii, then the clusters do not touch. If the distance is less than the sum, then they do touch. Using this definition, every cluster will also touch and form a connection with itself. The labels of each of the clusters forming the pairs are recorded in order to determine the clustered groups of clusters that all connect together.



(a)

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Cluster: 1 2 3 4 5 6 7 8 9 10 11 12

(b)

Figure 3.2: The system of clusters in (a) is used to construct an adjacency matrix in (b). In (b), a matrix element (p, q) is equal to 1 if cluster p touches cluster q. If $(p, q) = 1$, $(q, p) = 1$ by definition. As every cluster also touches itself, every element along the diagonal is equal to 1 by default. The number sitting under each column of the adjacency matrix is the label of the cluster that column represents.

The next stage in the grouping process is to construct an 'adjacency matrix' for clusters found to be within touching distance of each other. The adjacency matrix of the system can be easily generated from the list of touching pairs found previously. For a system of n clusters, the adjacency matrix associated with the system is a symmetric $n \times n$ matrix, where matrix element $(p, q) = 1$ when cluster p touches cluster q, and is equal to 0 otherwise. In order for the grouping to work, the clusters are also considered to touch themselves, so that the diagonal entries $(i, i) = 1$ by default. The adjacency matrix for a simple system is shown in figure 3.2, to demonstrate the connection between the cluster system and the matrix.

3.2.1 The Dulmage-Mendelsohn decomposition

A Dulmage-Mendelsohn (DM) decomposition is then applied to match clusters into 'well-determined' connected groups. The Dulmage-Mendelsohn decomposition process was first proposed by Dulmage and Mendelsohn [114], and was applied in a similar context by Pothen et al. [115]. The process described by Pothen et al. is applied here. In general, the Dulmage-Mendelsohn decomposition takes the adjacency matrix and permutes the rows and columns to form a block upper-triangular matrix, segregated into 'over-determined', 'under-determined' and 'well-determined' sections. However, as the systems we are discussing are always well-determined, we will only discuss the well-determined block. With

percolating systems, the rows and columns of the adjacency matrix are shifted so that the non-zero entries are gathered up into small blocks arranged along the diagonal. In practice, this means that all the entries from clusters that connect to each other as a group are shuffled together. The small, finely-decomposed blocks are square, symmetric and arranged along the diagonal of the decomposed matrix. More importantly, the Dulmage-Mendelsohn decomposition is performed so that each block of the decomposed matrix contains all of the connections that link together, and only those connections. Because of this, each block is associated with a group of connected clusters - for any particular block in the decomposed matrix, if we know which cluster is associated with each of the non-zero entries of the block, we know that those clusters must be connected, and so must form a group.

Extracting the grouping from this decomposed matrix requires knowledge of two things: the permutation ordering $\hat{\mathbf{p}}$ required along each axis to transform the adjacency matrix into the DM-decomposed matrix, and the location $\hat{\mathbf{l}}$ of the top-left-most entry of each small block. In our case, as the DM-decomposed matrix is also symmetric, the ordering is the same for both the rows and columns. Additionally, the location of the top-left-most entry of each block lies along the diagonal so that only the row or column index is required to locate it. Because of this, both $\hat{\mathbf{p}}$ and $\hat{\mathbf{l}}$ will be vectors. Crucially: the number of entries in $\hat{\mathbf{l}}$ tells us the number of groups in the system, as each entry $\hat{\mathbf{l}}(n)$ corresponds with the first entry of block n, and so can be said to correspond to group n. Thus, $\hat{\mathbf{l}}(1)$ corresponds to group 1, $\hat{\mathbf{l}}(2)$ corresponds to group 2 and so on. $\hat{\mathbf{p}}$ contains the labels of every particle in the system, but is arranged so that the entries between $\hat{\mathbf{l}}(n)$ and $\hat{\mathbf{l}}(n+1)-1$ correspond to the particles associated with the n^{th} block and thus located within the n^{th} group.

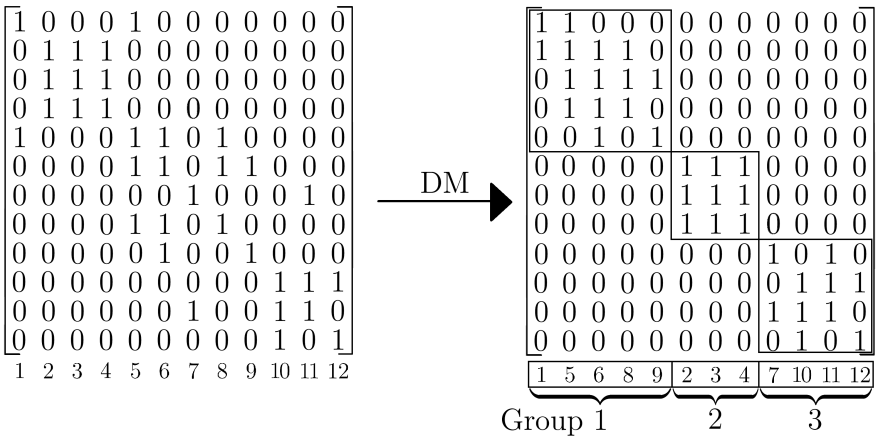


Figure 3.3: An illustration of the end result of a Dulmage-Mendelsohn (DM) decomposition. The adjacency matrix produced in figure 3.2 is processed by a DM decomposition algorithm to produce a block-diagonal matrix. The permutation vector detailing the re-ordering of the rows and columns required to obtain the block-diagonal matrix is located in the rectangles under the right-hand matrix - each small rectangle corresponds to the cluster labels associated with the non-zero 'connection' entries located in the matrix above it. The first set of labels are thus located in group 1, the next set in group 2 and so on.

Figure 3.3 gives a brief example, using the adjacency matrix constructed in figure 3.2. Each row and column of the adjacency matrix is permuted in order to produce the decomposed matrix, and the column labels located below the matrices have been permuted accordingly. Following the decomposition, the non-zero entries have been grouped into three blocks - a 5x5 block, a 3x3 block and a 4x4 block. We can see that the top-left-most entry of the first block is located at (1,1), the top-left-most entry of the second block is located at (6,6) and the top-left-most entry of the final block is located at (9,9). The first-entry vector $\hat{\mathbf{l}}$ will only contain the row index of each of these, so that $\hat{\mathbf{l}} = [169]$ in this case. The permutation ordering vector is provided below the DM matrix, and simply contains the order in which the rows/columns were placed - in this case, it is

$$[1 \quad 5 \quad 6 \quad 8 \quad 9 \quad 2 \quad 3 \quad 4 \quad 7 \quad 10 \quad 11 \quad 12] \tag{3.3}$$

Using our rule from before, we see that the cluster labels associated with group n are found by looking at entries $\hat{\mathbf{l}}(n)$ to $\hat{\mathbf{l}}(n+1) - 1$ in the ordering vector. In our case, this means that group 1 contains clusters located at positions 1 to 5 in the ordering vector, which happen to be clusters 1, 5, 6, 8, 9. Group 2 contains the clusters located at positions 6 to 8 (clusters 2, 3 and 4). Group 3 contains the clusters located between position 9 and the end - these are clusters 7, 10, 11 and 12. These groupings correspond with the grouping that can be readily observed by simply looking at the simple particle system in figure 3.2.

Following the DM decomposition, we can assign the group number to each of the clusters associated with that group in a new 'group vector', where the value at index i is the group number associated with cluster i. The group number of each cluster is saved in this vector, and the number of clusters assigned to each group number is saved in a vector of group strengths. By the end of this step, the system is very similar in appearance to the systems represented in figure 3.4, with discs representing the clusters, and the colour of the discs indicating the immediate group they belong to. Networks with spanning groups will resemble those in figure 3.5.

3.3 Conductance of individual connections

The conductance assignments between groups are performed by first making use of Delaunay Triangulation [35], mentioned in sections 2.3 and 2.5.3, to construct connections between nearby cluster centres. The edges of each of the triangles generated by the Delaunay triangulation are saved in a list. For every group in the system, the list of edges belonging to the Delaunay triangles is examined for edges that involve exactly one cluster associated with the group in question, and the edges which meet this criterion are stored.

The lengths of the edges found above are compared with each other if they connect the same two groups. As the program at this stage can only handle tunnelling conductances, only the shortest of the edges between any two groups is kept, as the shortest distance will provide the best tunnelling path due to the exponential decay of the tunnelling rate as the distance increases. The shortest

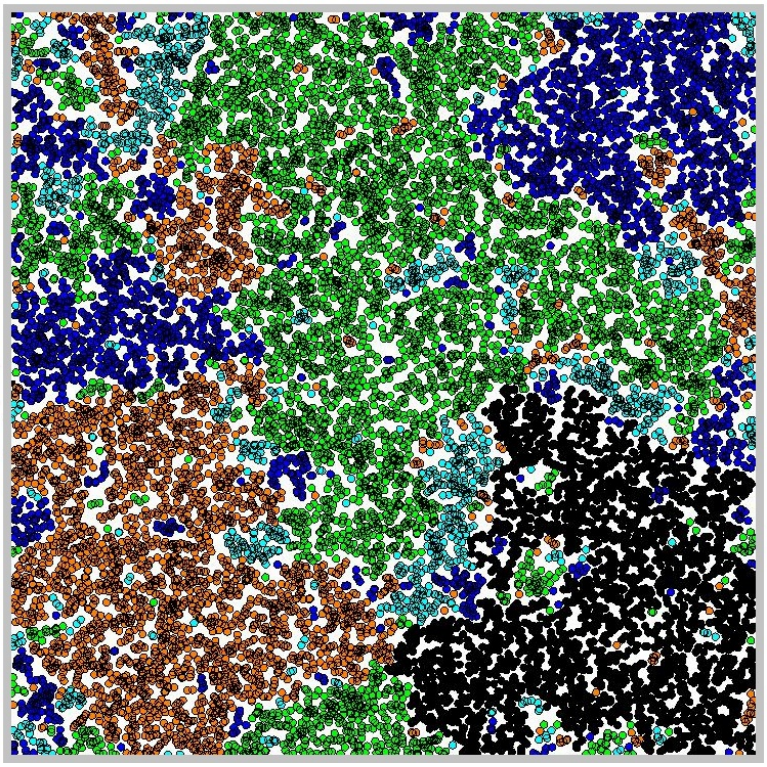


Figure 3.4: A tunnelling network following deposition and grouping in a 100x100 system. This sample has a coverage of $p = 0.68$, making it one of the few samples which does not span even above p_c at this system size. Clusters belonging to the same group have been given the same colour, and transfer current between each other without resistance. Only the largest group is coloured black. This system is only capable of transporting current between terminals via tunnelling connections between differing groups.

path between the two groups is assigned a tunneling conductance related to the distance between the clusters involved: $\sigma = e^{-\beta|z_1-z_2-(r_1+r_2)|}$. The expression $\mid z_1 - z_2 - (r_1 + r_2) \mid$ gives us the distance between the edges of two clusters 1 and 2, located at positions z_1 and z_2 and with radii r_1 and r_2. As mentioned previously, this is performed for every connection involving every group. The conductances generated between pairs of groups are saved in a conductance matrix. The entry at position (i, j) in the conductance matrix describes the conductivity of the connection between groups i and j. This matrix of conductances provides the basis for a random resistor network. At this point, the system of superconducting groups and resistive tunnelling connections is interpreted as a random resistor network, with only the tunnelling connections resisting the flow of current between the terminals. This is presented in figure 3.6, where the black links represent the superconducting groups, and only the red tunnelling connections act as resistors.

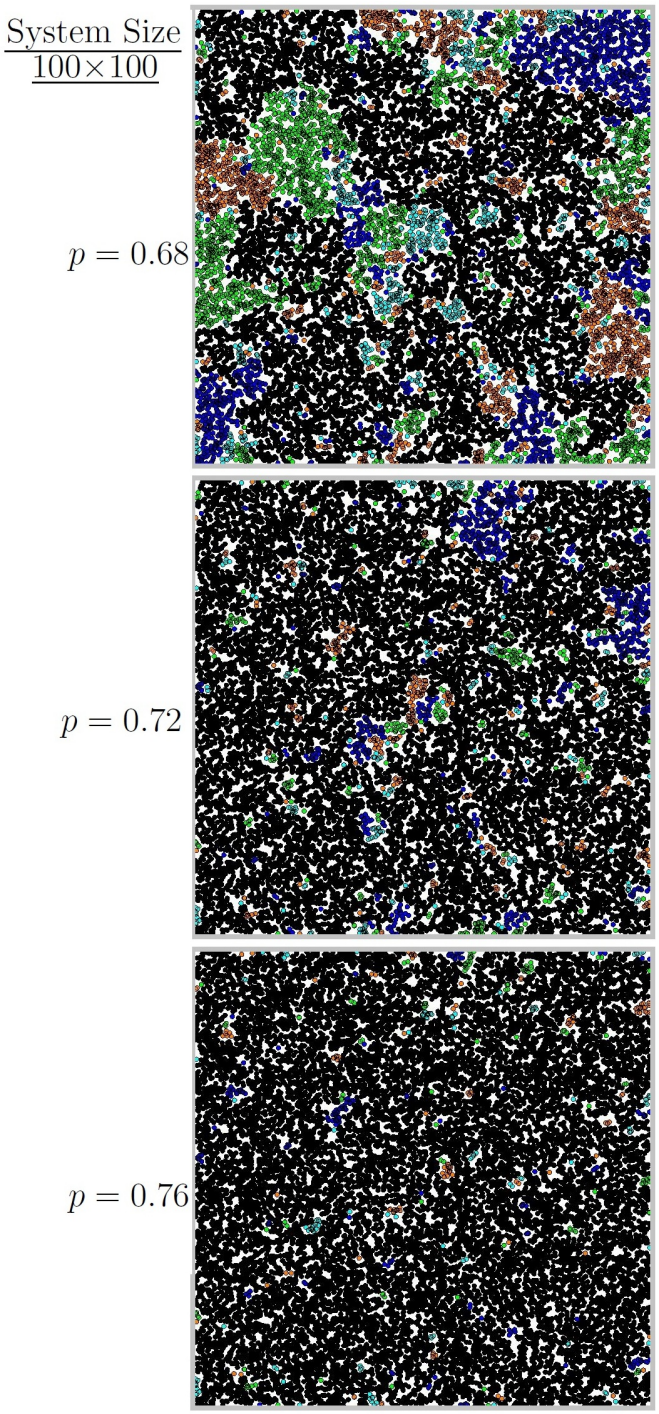


Figure 3.5: Three representative spanning networks following deposition and grouping in a 100x100 system. All three networks belong to the same sample, at coverages of $p = 0.68$,0.72 and 0.76 respectively - all of which are above p_c. Clusters belonging to the same group have been given the same colour, and transfer current between each other without resistance. Only the largest group, which spans the system at these coverages, is coloured black. Without tunnelling, only the spanning groups are able to transport current.

3.4 Conductance of the network

The final step is to calculate the conductance of the entire system. The groups that do not connect with the groups that allow them to indirectly reach both

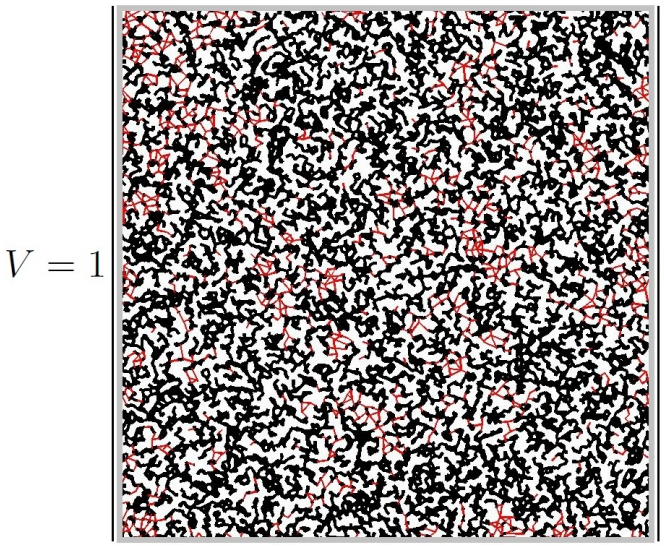


Figure 3.6: A tunnelling network in a 100x100 system at $p = 0.68$. The superconducting connections between particles of the same group are given by the black lines, while the shortest distances between differing groups are the most likely place for current to tunnel between the groups, and so are assigned a resistance based on the shortest distance between them. These tunnelling connections are shown in red.

terminals of the system are removed, as they will be unable to have any current passed to them. As discussed in section 2.5.2, the idea is to find the voltages of each group in the system, and the current flowing through the system, by solving Kirchoff's laws. To do this, we need to solve equation 2.20 for every group in the system, as every group acts as a node in the network. In order to achieve this, a final 'G-matrix' must be constructed.

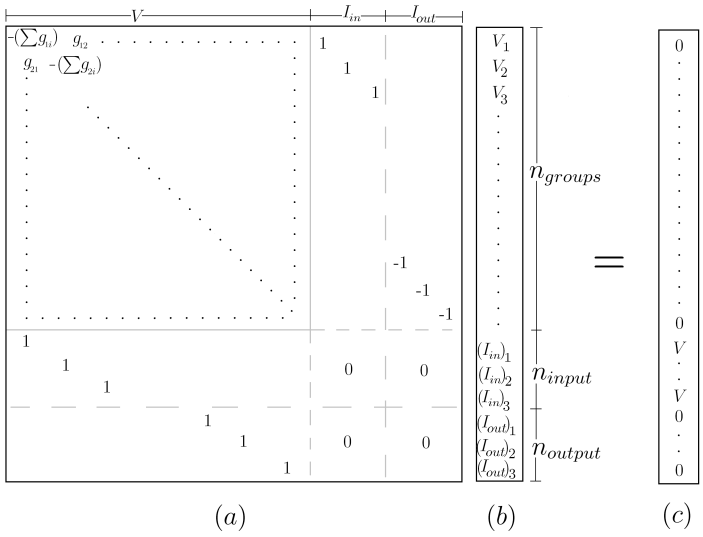


Figure 3.7: The final matrices involved in calculating the voltages of each group and the currents flowing into and out of the terminal groups.

The G-matrix contains almost all of the information to solve Kirchoff's laws for the system, as well as to set the voltage/current boundary conditions for the

system. From equation 2.20, we can see that for any individual group i, the current flowing into the group must be equal to the current flowing out of the group, so that

$$(I_{in})_i - (I_{out})_i + \sum_j g_{ij}(V_j - V_i) = 0 \tag{3.4}$$

Where I_{in} is only non-zero if group i touches the input terminal, and I_{out} is only non-zero if group i touches the output terminal. The above equation can be rearranged to form

$$\sum_j (g_{ij}V_j) - \sum_j (g_{ij}V_i) + (I_{in})_i - (I_{out})_i = 0 \tag{3.5}$$

This equation forms the basis of the final G-matrix.

We also need boundary conditions. The groups in contact with the input terminal should share the voltage value of that terminal, relative to the output terminal. If the voltage between the input and output terminals is V, then for every group j in the system, $V_j = V$ if group j touches the input terminal. Alternatively, if group j is in contact with the output terminal, its voltage should be $V_j = 0$.

This is enough information to find the voltage of every group in the system relative to the output terminal, and to find the currents entering the input terminal groups and leaving the output terminal groups. The G-matrix $\boldsymbol{G}$ is constructed as shown by figure 3.7(a), whereas the vector of unknown voltages and currents $\hat{x}$ is assembled in (b), and the vector of required equalities $\hat{v}$ is located in (c). (a) presents $\boldsymbol{G}$, where the top n_{groups} rows are given by equation 3.5. Row i features a 1 in one of the I_{in} columns if group i touches the input terminal. Similarly, row i features a -1 in one of the I_{out} columns if group i touches the output terminal. The following n_{input} rows set the voltage for the input terminal groups to V - there is one row for every group in contact with the input terminal, and entry j along that row equals 1 if group j touches the input terminal. The final set of n_{output} rows sets the voltage of the output terminal groups to 0 - there is one row for every group in contact with the output terminal, and entry j along that row equals 1 if group j touches the output terminal. (b) is the vector of unknown values $\hat{x}$ that we are hoping to solve for. There are n_{groups} unknown voltages, n_{in} unknown input currents, and n_{out} unknown output currents. (c) presents the vector of required equalities $\hat{v}$ - these are all equal to zero, aside from the input terminal voltages located in positions $n_{groups} + 1$ to $n_{groups} + n_{in}$, which are equal to V.

Finally, we have $\boldsymbol{G}\hat{x} = \hat{v}$. The solution vector is produced by solving $\hat{x} = G^{-1}\hat{v}$ using standard linear algebra techniques. This solution provides the voltages assigned to each group and the current flowing through the groups situated along each terminal. The sum of the input currents, divided by the voltage difference between the two terminals, gives us the conductivity of the system.

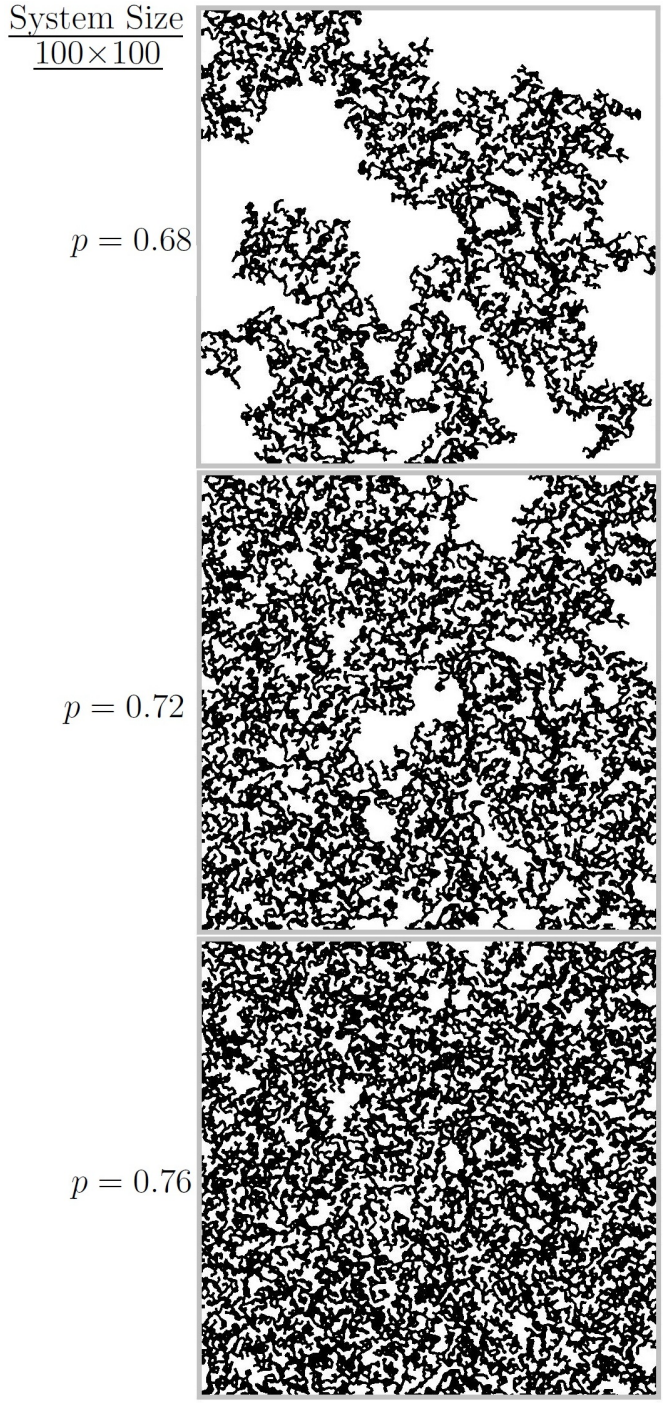


Figure 3.8: The same three networks from figure 3.5 with only the percolating backbone shown. Each black line represent a connection between clusters. Without the introduction of weak links, it would not be possible to calculate the resistance of any of these networks, as they all span the left and right-hand sides of the system.

3.5 Introducing weak links

While the approach discussed in this chapter works perfectly well for tunnelling systems, it encounters errors or crashes when any of the groups spans between the two terminals, as would be expected for coverages $p > p_c$. As groups are con-

sidered to be superconducting, there should be no resistance for current traveling within groups. This situation is the equivalent of describing the groups as nodes in a random resistor network. Given this, the concept of voltage difference only makes sense when applied to the resistive links between groups. The problem occurs when there are no resistive links separating the two terminals, a situation which exists when a group spans and touches both terminals. The initial conditions above attempt to assign both a voltage of V and 0 to this same group - an obvious contradiction! Any spanning path that has no resistive or tunneling links somewhere along it will expose this problem where the contradiction embedded in the network will prevent Kirchoff's laws from being solved to retrieve the conductivity or voltages. It is possible to screen samples with spanning groups out of any attempts to measure the conductivity, but that makes any attempts to numerically explore samples with coverages above p_c nigh-impossible. Figure 3.8 provides instances of systems which are unsolvable using the unadjusted method discussed here, due to groups spanning between the terminals.

Introducing the properties of a certain physical feature to the system may not only improve the relationship between the simulation and the physical scenario, it may also solve the simulation problems in one step! The concept of 'weak links' in the context of superconductors has been an active area of study for decades, particularly as an extension to the theory of Josephson junctions [53]. In the continuum percolation systems discussed here, it is possible for the overlap between clusters to be very small and tenuous. As discussed by Tinkham [45], Josephson junctions can be formed when a constriction of superconducting material exists between two larger superconducting bodies if the diameter of the constriction is less than the superconducting coherence length $\xi_s(T)$. If the overlaps between clusters satisfy this, they may qualify as Josephson junctions, and so constitute 'weak links' in the system. These weak links are also expected to be the network connections with the lowest critical currents, and so should dictate the critical current of the system. By appropriately selecting weak links to fragment the spanning groups into multiple segments, we can not only compute the more physical reactions of the groups themselves to an external voltage, we can also avoid the problems associated with spanning groups! An issue remains that by including too many connections, the ensuing network will be too complex and time-consuming to investigate within a reasonable period of time. With that in mind, it is worth investigating various methods of selecting the weak links, in an attempt to balance the accuracy of the computations with the time requirements of the program.

The goal of this thesis is to extend the work done on the previous program so that it is able to produce the expected properties and power-laws of continuum percolation systems that are prepared in either the superconducting or normal states. Various approaches, chosen to reduce the time taken for this extended analysis, will be compared based on their ability to replicate known properties and the time taken to do so, so that an appropriate model can be chosen for future research. The remainder of this thesis will seek to introduce the concept of weak links in a manner amenable to simulation, so that 'weak-link' connections can be found and used to compute the conductivity and V-I properties of percolating systems when $p > p_c$. Chapter 4 will discuss the method of selecting weak

links, and provide three models for examination. It will also detail an additional algorithm used to simulate the systematic increase or decrease of current flowing through the network, and the effect of this on the states of individual connections. This algorithm is called the 'current ramp'. Chapter 5 will present and discuss the results associated with the three models, quantified by their system properties, conductivities and reactions to changing current.

Chapter 4

Weak-link connections

4.0.1 How do we find weak links?

The goal is to find the few connections in the percolating superconducting system that are the most likely to surpass their local critical current I_c when some current I is passed through the system. In a continuum percolation system, the clusters that are added to the system occupy random positions and form a random network of nodes, links and blobs, as discussed in sections 2.1 and 2.3. A key point regarding the nodes-links-blobs picture is that the connections within blobs vastly outnumber the links connecting any two blobs. Parallel sets of these links provide fewer alternative paths for current than the interior of the blobs they connect together. This implies that the parallel links are less able to redistribute the current, in order to avoid exceeding their collective I_c's, than the blobs themselves. As the current I is forced higher, the lack of parallel paths will eventually drive enough current through these chokepoint-like links that locally $I > I_c$, turning them normal. From this argument, we can expect that the links separating blobs are the most likely to go normal as the current in the system increases, and the conductivity of the sample will be influenced by the state of the links. Additionally, as any input current would be distributed over a larger number of connections, the conductance of a blob should be much higher than that of any adjacent weak link leading into it. It is expected that the resistance of the sample is dominated by the links connecting the blobs together, in both the superconducting and normal systems. These links are the weak links that our search will be attempting to find.

There exist a few ways in which we might identify necks as being potential weak links. The most extreme method is to claim that every link is weak, so that every connection between every cluster pair in the system is a possible weak link. This is the brute force approach, which examines the connections from all clusters to all other clusters. By examining every connection available in this way, we effectively allow every inter-cluster connection in the system to go to the normal state individually. With every connection in the normal state, the system exists as a random resistor network, where the number of resistive connections increases as the coverage p increases. Modeling every connection as a potential weak link is the most likely approach to replicate the properties of real samples,

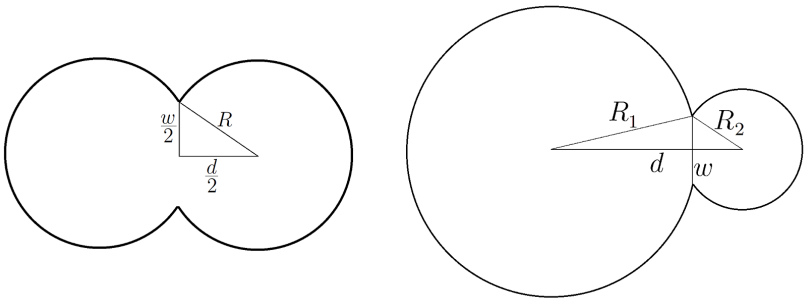


Figure 4.1: The neck width w associated with clusters of equal radii R, as seen in Eq. 4.1, is constructed on the left. The neck width w associated with random cluster radii R_1 and R_2, from Eq 4.2, is constructed on the right.

particularly for high currents, temperatures or incident magnetic fields. Within our simulation, however, negotiating the vast web of available connections is computationally intensive. This method is also the most time-consuming, due to the sheer time required to analyse networks constructed out of every single connection in the system. The exact difference in time taken between the 'All to All' (AtA) method and its alternatives will be discussed in section 5.6 of the results.

An alternative, rougher approach is to only select appropriately weak links which sit outside of blobs. An average L=200 simulation at $p = 0.68$ consists of 60,000 clusters, with roughly a quarter of a million connections between pairs of overlapping clusters. Each of these connections is modeled as a wire, with a cross-section determined by the width of the 'neck' created by the cluster overlap as seen in Figure 4.1. The connections with the lowest I_c will be the wires with the smallest cross-section, corresponding to necks that are very narrow. However, if these narrow wires sit in the middle of a blob, the multitude of connections within the blob can easily redistribute the current and mitigate the weakness of the narrow wires located inside of the blob. Because of this, it is sensible to not only select necks that are particularly narrow, but also to avoid choosing necks that exist inside blobs. This argument suggests that we should be able to approximate the AtA approach by only accepting the weak links that sit outside of blobs. In order to accomplish this, two issues must be contended with:

1. How to define 'narrow necks'
2. How to select only the weak links between blobs

4.0.2 Defining weak links

Firstly, we can define the narrow necks, and thus the weak links, in a geometric fashion: we consider the connection to be a weak link when the chord of the overlap (the neck) between the clusters is smaller than a certain threshold value. The neck width w for the overlap of two clusters separated by a distance d can be determined for a system of clusters with identical radii R by a Pythagorean

argument, where (from Figure 4.1)

$$w^2 = (2R)^2 - d^2 \tag{4.1}$$

When clusters are allowed to have random radii, we must use a more complete formula for the neck width. For two clusters 1 and 2 (from figure 4.1), the formula is

$$w = \frac{1}{d}\sqrt{-(d+R_1+R_2)(d-R_1-R_2)(d+R_1-R_2)(d-R_1+R_2)} \tag{4.2}$$

which reduces to equation 4.1 when $R_1 = R_2$ as follows:

$$\begin{aligned} w &= \frac{1}{d}\sqrt{-(d+R_1+R_2)(d-R_1-R_2)(d+R_1-R_2)(d-R_1+R_2)} \\ &= \frac{1}{d}\sqrt{-(d+2R)(d-2R)(d)(d)} \\ &= \frac{\sqrt{d^2}}{d}\sqrt{(2R+d)(2R-d)} \\ &= \sqrt{(2R)^2 - d^2} \\ \rightarrow w^2 &= (2R)^2 - d^2 = \text{ Equation 4.1} \end{aligned}$$

The neck width is uniquely determined by the distance between the clusters. Fortunately, the distance between the clusters is much easier for the computer to find directly than the neck width! If we remember the location of any cluster as a complex number z, where $z = x + iy$, then the distance between any two clusters is simply

$$d = \mid z_{12} \mid = \mid z_1 - z_2 \mid$$

The threshold value is the largest neck width that were willing to consider as a potential weak link. We can use it to find the threshold distance (with respect to the cluster diameter) that can separate two weakly linked clusters by rearranging equation 4.1

$$d_0 = \sqrt{(2R)^2 - w_0^2} \tag{4.3}$$

Both equations 4.1 and 4.3 will be used to produce the threshold value which determines whether links will be considered weak or strong.

4.0.3 Regrouping with smaller radii

Secondly, as discussed at the end of section 4.0.1, we should attempt to exclude intra-blob necks. We require an algorithm that excludes links in the middle of blobs, and only selects connections that divide a group into two or more when severed. From our discussion of the interconnectivity of blobs, we should expect that if superconducting links surround a normal one, current will preferentially avoid the resistive normal connection and the weak link connection will be irrelevant. Knowing this, we should concentrate our attention on distinguishing

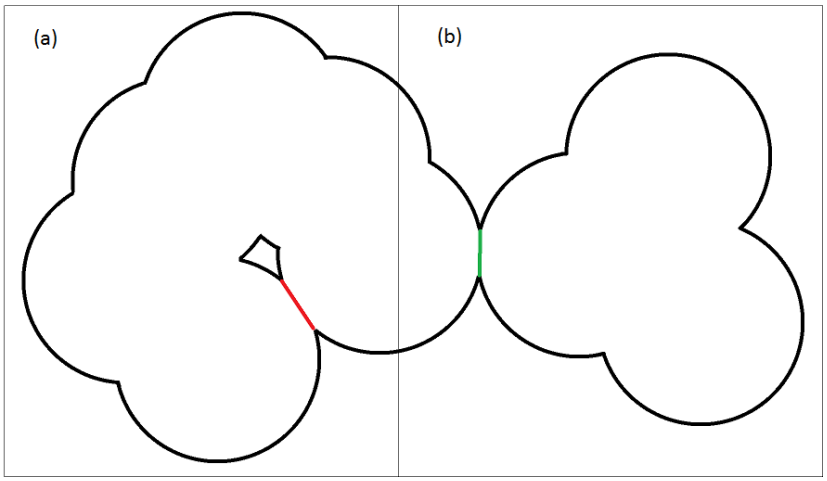


Figure 4.2: Two connections, shown in red in (a) and green in (b), have neck widths less than the threshold. However, (a) is contained within a blob and so is discarded. The green link in (b) will separate the two blobs if it is removed, and so is an important weak link.

between links around the boundaries of blobs and those in the middle of blobs. A large number of links meeting the threshold requirement might be situated in the middle of blobs, where alternative routes can help the weak link remain superconducting until the entire blob goes normal (see Figure 4.2(a)). Recall from section 2.2.1 that the connection is considered to be a link if it fragments the group of clusters that the connection belongs to into multiple groups when the connection involved is removed (see Figure 4.2(b)). If there are several links which separate the same two blobs, each meeting the weak-link threshold criteria, they still qualify as potential weak links as removing all of them would split only those two groups apart. The approach that has been taken here has two steps. The first step is to break or remove every connection that satisfies the proposed threshold criteria. The second step is to regroup the particles using only the remaining connections and to record every broken connection as a weak-link and a potential resistor in the network.

The important first step is to systematically remove the weak-link connections that satisfy our geometric 'neck width threshold'. Given a particular threshold separation distance d_0 from equation 4.3, we will construct a 'radius reduction factor' λ, which multiplies the radius of every cluster in the system to make them smaller. This approach is similar to previous examinations of disks with hard cores [14]. If we assume that the cluster radii are chosen randomly from an arbitrary distribution, we can find the value of λ for a particular average cluster radius $\langle r \rangle$(Fig. 4.3(a)). λ is chosen so that

$$d_0 = 2 \langle r \rangle \lambda \tag{4.4}$$

From this we can see that $\lambda = \frac{d_0}{2\langle r \rangle}$. Clusters whose centers are exactly a threshold distance apart will be only just touching following the transformation. Clusters that are closer together than the threshold will still overlap, and are considered to be strongly linked together(Fig. 4.3(b)), and clusters that are further apart will no longer touch and are considered to form a weak link(Fig. 4.3(c)). It is possible

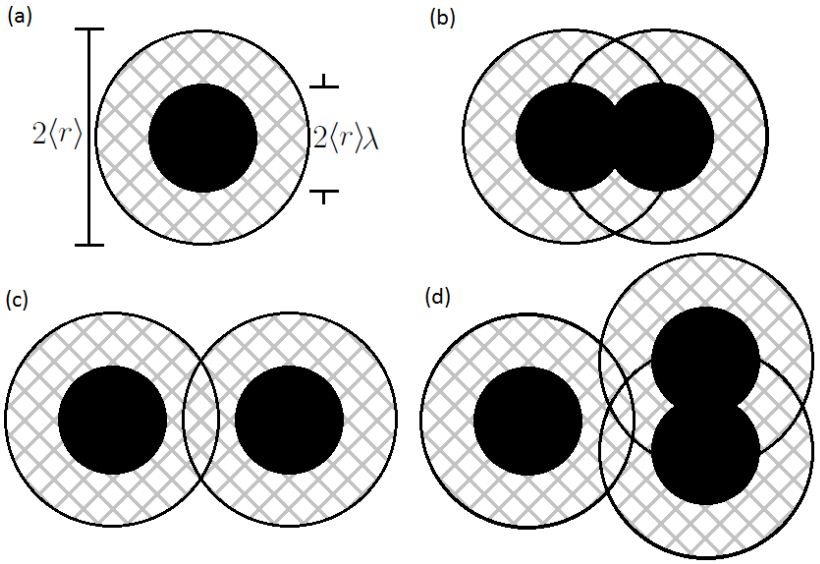


Figure 4.3: An illustration of the areas covered by particles before (in grey) and after (black) the radius reduction factor is applied. (a) An individual disk with diameters before ($2\langle r\rangle$) and after transformation ($2\langle r\rangle\lambda$) indicated. Here, $\lambda \approx 0.5$. (b) Disks that remain in contact following the transformation are considered strongly linked. (c) Disks that separate following the transformation are considered to be weakly linked and have a conductance attached to them. (d) In practice, multiple weak links can connect groups together.

to sever multiple connections involving the same cluster and thus produce several weak link connections between two groups(Fig. 4.3(d)).

The second step is to recollect all the clusters that continue to touch each other into new groups. The reduction of the cluster radii by a factor of λ will fragment the group they originally belonged to into several new groups. Of course, we can re-use the original grouping algorithm discussed in section 3.2, where clusters were grouped together by constructing an adjacency matrix for clusters found to be within touching distance of each other, then using a Dulmage-Mendelsohn Decomposition to uncover the groups that each cluster belonged to [115]. The purpose of this is to work out which of the new groups each weak-link joins together, so that we can construct an appropriate 'G-matrix' to find the conductivity of the system.

Once the new groups are formed, the method of finding the weak links is similar to the method of setting the tunnelling connections described in section 3.3. Again, a Delaunay Triangulation is constructed using the cluster centers as points. Again, for each group, only the Delaunay triangle edges connecting the group in question to other groups are investigated. However, this time the edges are considered to be 'weak-link' connections if the distance d between the clusters involved is small enough that they were previously touching (so that $d < 2r$), but is big enough that they do not touch following the radius reduction process (so that $d > 2r\lambda$). The edges satisfying these two requirements represent the connections between weakly-linked clusters. The new groups that they connect together, as well as the distances d involved, are saved. As in 3.4, a new conductivity matrix is created

where entry (i, j) is non-zero if new group i is connected by a weak link to new group j, and the conductivity of this weak-link is proportional to the neck width w, calculated from the separation distance d using equation 4.1. It is worth noting that the new groups themselves are not resistive - only the weak links connecting the new groups together are able to present resistance to the flow of current.

In summary, the method for finding weak links is as follows:

- Choose a threshold distance d_0
- Calculate the multiplicative factor λ using Eq. 4.4. We call λ the radius reduction factor.
- Multiply all cluster radii by λ
- With the reduced radii applied to all clusters, weak-link connections will be broken because they will have been further apart than the threshold, but strong links will remain. Find the groups again using the Dulmage-Mendelsohn Decomposition.
- Where groups fragment into smaller groups, we can look for the connections between the new groups that correspond with the removed, previously-touching weak connections and assign a conductance to them. Weak link found!

4.0.4 The chosen models

Two approaches have been explored regarding the application of a radius reduction value to each coverage. The first was to maintain a constant radius reduction factor (CRR), regardless of coverage. This model assumes that whether or not a particular neck should be considered as a weak link is purely governed by the geometry of the connection, and that this geometrical consideration is independent of the coverage. This is a reasonable assumption when the incident current, incident magnetic field and ambient temperature are held constant, but it allows the possibility that groups will continue to span the system at high coverages. If groups continue to span the system, the conductivity of the system cannot be calculated due to the problems mentioned in section 3.5, and so radius reduction factors that allow the possibility of spanning groups should be avoided. The value of the radius reduction factor λ used for the CRR model is $\lambda = 0.85$. This choice of λ was to ensure that the system would avoid spanning for the range of p being investigated - $p = 0.68$ to 0.8 - without significantly slowing down the model by considering too many connections at once.

We can derive the 'reduced coverage' p_r produced by the radius reduction process

from equation 2.12:

(4.5)

Before $(p_{total}) = -\ln(1-p)$, afterwards $(p_{total})_r = -ln(1-p_r)$

But $(p_{total})_r = \dfrac{na_r}{A}, \quad a_r = \pi\,(r\lambda)^2 = \lambda^2\pi r^2 = \lambda^2 a$

So $(p_{total})_r = \lambda^2\dfrac{na}{A} = \lambda^2(p_{total})$

Substituting in ln relationship: $-\ln(1-p_r) = -\lambda^2\ln(1-p)$

Therefore, $p_r(p,\lambda) = 1-(1-p)^{\lambda^2}$ (4.6)

The second approach was to choose λ so that a constant reduced coverage p_r (CPR) of the transformed sample was maintained. This model forgoes any assumptions about the environment, and instead attempts to guarantee that there will not be a spanning group between the terminals. This implies that the system can potentially demonstrate resistive properties at any coverage. If the chosen apparent coverage was p_r, then at $p = p_r$ the λ used would be 1. At $p > p_r$, λ would be less than 1, and would be constructed so that the first layer coverage following the transformation would be equal to p_r.

The radius reduction factor λ for the p_r approach is found by rearranging equation 4.5:

$$-\ln(1-p_r) = -\lambda^2\ln(1-p)$$

$$\rightarrow \lambda(p,p_r) = \sqrt{\frac{\ln(1-p_r)}{\ln(1-p)}} \tag{4.7}$$

(4.8)

This gives us the radius reduction value required to produce a system of desired coverage p_r from a given input coverage p. The radius reduction value generated for each coverage in this manner defines the threshold value for weak links for that coverage. We used $p_r = 0.65$ for the majority of the trials that use the CPR model. As $p = 0.65$ is just below the critical percolation threshold $p_c = 0.676339$, we would avoid spanning groups while still being close to p_c. Additionally, if p_r was set too small, too many connections would be considered at high coverages and the program would slow down significantly.

Both models have been compared with the 'all clusters to all clusters' (AtA) model which is expected to demonstrate equations 2.3 and 2.19 as it matches the canonical inverse Swiss cheese model closely. The conductivity distributions and other statistics associated with the CRR and CPR models are not guaranteed to align with the AtA model, and so the results for CRR and CPR will only make sense when compared with those of a reasonably reliable control case.

4.1 Ramping the current

Once the new network of weak links is established, it becomes possible to decide whether links are normal or superconducting by passing current through the net-

work, then comparing the current through each weak link with its own critical current. Connections with a current passing through them that exceeds their critical current will go normal - that is, their resistance will increase towards the maximum resistance of the connection. On the other hand, if the current going through a connection is less than its critical current, it will switch into a superconducting state. An idealised superconducting connection would possess zero resistance. In simulations, however, attempting to set the resistance to $r = 0$ (the same as setting the conductance $g = \infty$) causes significant problems with computation. MATLAB attempts to compute a direct solution to Kirchoff's equations using the values provided. If one of the values given for the conductivity of the connections is infinite, then MATLAB will immediately stop and present an error message, to avoid attempting to calculate the value of infinity, regardless of whether a tenable solution for the entire network exists. Because of this, conductances must be given as finite numbers for both normal and superconducting connections. By making the superconducting conductance value g_{sc} several orders of magnitude larger than the largest normal conductance $\max(g_n)$, solutions can still be calculated while minimising the impact of the non-zero superconductor resistance value. In the simulations performed for this thesis, $g_{sc} = 10^4$ units.

In order to ramp the current through a particular network of weak links, several things must be established. Firstly, the initial current entering the system must be chosen. In the event of ramping the current down towards 0, the value of the initial current is chosen so that every possible connection has enough current flowing through it to remain normal. The current required for all of the involved connections to go normal should also be expected to increase in proportion with system size. Dangling bonds are not expected to carry current and so can be left out of consideration. Exactly what current is required for the backbone to be fully normal is itself difficult to resolve analytically and so, in this case, the initial current is chosen to be an arbitrarily large value. An initial current of 10^6 units has been adequate for 200×200 and 400×400 systems, although the dependence on system size has not been investigated.

Secondly, the critical current I_c of each individual connection must be constructed. As discussed in section 2.4.1, the universality of the exponent v present in equation 2.17 is dependent on whether the constriction size, or in our case the neck width w, is greater or smaller than the superconducting coherence length $\xi_s(T)$. The non-universality of the critical current exponent is important to the simulation, as the size of the necks relative to the superconducting coherence length ξ_s is now a necessary factor in determining the evolution of the system's I_c as the coverage increases. Because of this, the relationship between neck width and $\xi_s(T)$ needs to be explicitly defined in the program if it is to reflect the properties of the experimental system. With this in mind, as each connection is assigned a critical current, the decision must be made to give the connection either a Josephson-like critical current, or a depairing-like critical current. In the case of this thesis, only the depairing case, where $w > \xi_s(T)$ will be investigated.

In practice, the assignment is performed by producing a $n_{groups} \times n_{groups}$ matrix to store the critical currents associated with the connections connecting each weakly-linked group to its neighbours. This matrix is almost the same as the conductivity matrix constructed by the conductance assignment performed in

section 3.4. For any filled entry (i, j), corresponding to the connection between groups i and j, the critical current I_c is either set to 1 in the Josephson case, or set proportional to the neck size w of the connection. In this simulation, all necks are assumed to simultaneously be larger or smaller than $\xi_s(T)$ and so all are given the same type of critical current.

4.1.1 Altering the G-matrix

Once the critical currents have been set, the algorithm proceeds in the same way as the usual Kirchoff solution from section 3.4. Every group that does not participate in the transport of current following the weak-link regrouping is removed from the system, and the G-matrix to be used for the first solution iteration at each current is prepared, where the conductivity of each connection is equal to the normal resistance of the weak-link, $g = w$. One significant difference between the G-matrix used for the current ramp, and the one used to find the fully-normal system conductivity in section 3.4 is that an additional group is added to the system to represent the input terminal. This is due to the fact that the initial condition that is set during the current ramp is the input current. While the voltage of each group attached to the input terminal can be easily set to V, as was done in section 3.4, the current to be sent to each group cannot be easily allocated without knowing how much current each group should receive to begin with. This issue can be tackled by allocating a total amount of current I_{in} that is sent to the terminal itself, to be distributed between the groups touching the terminal during the solution of Kirchoff's laws. The newly added terminal group is treated like every other group, with a permanently-superconducting link established between it and each of the groups in contact with it. In addition, terms that were previously used to set the terminal group voltages have been removed. They are replaced by a single term that is designed to set the input current I_{in} equal to the desired current I. The final G-matrix,the vector of group voltages that we wish to solve for and the vector of initial conditions, are shown in figure 4.4, with (a) showing the G-matrix in use for the current ramping process, (b) the vector of voltages, and (c) the vector of initial conditions. The system of equations represented by figure 4.4 is solved using standard linear algebra techniques whenever it is called.

4.1.2 Changing weak-link states

Once the altered G-matrix is established, the ramping process can begin in earnest, starting with the initial current. For the first, initial current, the system of equations shown in figure 4.4 is solved repeatedly over a number of iterations. At each iteration step, the system of equations is solved to find the voltage at every group. For every pair (i, j) of touching groups, the difference in voltage $|V_i - V_j|$ is worked out. Given Ohm's law, where $Vg = I$, we can compute the current flowing between the two groups i and j by multiplying the conductivity g_{ij} of the weak-link joining them, so that the local current $I_{ij} = |V_i - V_j| \times g_{ij}$. This local current is compared with the local critical current for the connection,

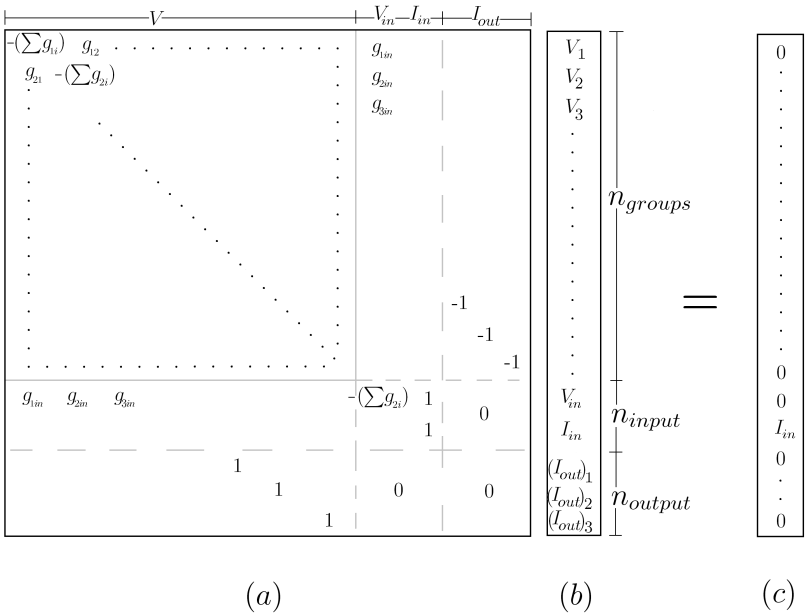


Figure 4.4: The final matrices involved in calculating the voltages of each group and the currents flowing into and out of the terminal groups during the current ramp process.

$(I_c)_{ij}$. If $I_{ij} < I_c$, the weak-link is considered to be superconducting, and so the conductivity of the link is set equal to $g_{ij} = g_{sc} = 10^4$ for the next iteration. If $I_{ij} > I_c$, however, the weak-link is considered to be normal. In this case, the weak-link is treated like the overdamped Josephson junctions discussed in section 2.4.1 where the conductivity for the next iteration is found using equation 2.16, so that

$$g_{ij} = R_{JJ}^{-1} = g_{ij}^{norm}\left(\sqrt{1 - \left(\frac{I_c}{I_{ij}}\right)^2}\right)^{-1} \tag{4.9}$$

where $g_{ij}^{norm} = w$ is the fully-normal conductivity of the connection between groups i and j, and is proportional to the neck width w separating them. After setting the conductances of the connections at the end of each iteration, these values are entered into the appropriate places in the G-matrix for the next iteration, and the cycle begins again. The solution will continue to iterate until every connection reaches equilibrium and stops switching between its normal and superconducting values, or until the 30^{th} iteration is completed. The voltage across the system V_{in} is saved at the end of every iteration.

4.1.3 Selecting the next current

At this point, the next current must be chosen for the ramping process to continue forward. In order to choose the next current, the ratio of local current to critical current, $\frac{I_{ij}}{I_c}$, for the connections between every pair of groups is collected and sorted into a list, arranged in ascending order. Entries where $\frac{I_{ij}}{I_c} < 1$ are discarded. The next current is chosen in an attempt to force at least one of the connections to go superconducting. Two options for this are used: a low-resolution choice, where the $\frac{I_{ij}}{I_c}$ value a tenth of the way down the list is chosen;

or a high-resolution choice, where only the top-most $\frac{I_{ij}}{I_c}$ value is chosen. The chosen value of $\frac{I_{ij}}{I_c}$ is the current-reduction value, λ_I. The next current, I_n, is then derived from the previously-used current, I_{n-1}, using

$$I_n = \frac{I_{n-1}}{\lambda_I} \tag{4.10}$$

This new current I_n is then placed into the current position in the vector of inital conditions, as seen in figure 4.4(c). The system conductivity for the previous current is calculated as $G = \frac{I_{n-1}}{V}$, where V is given by the voltage measured across the system during the final iteration at $I = I_{n-1}$ if the system reached equilibrium. Otherwise, V is equal to the average voltage of the final two iterations. The old current I_{n-1}, as well as the system conductivity G, are saved.

The current selection, then the iterated solving and readjustment of connection conductivities, is also repeated until the list of normal connections (connections ij where $\frac{I_{ij}}{I_c} > 1$) runs out. When this happens, the currents used, voltages across the system and the final conductivities at each current are saved. The general process is summarised in the flow-chart presented in figure 4.5.

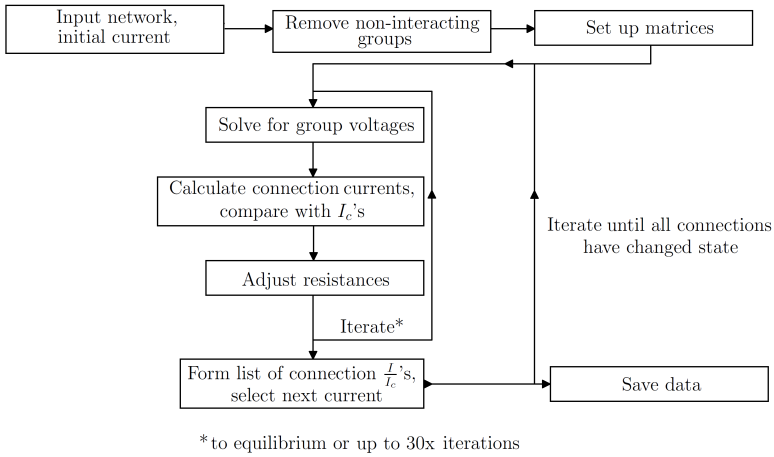


Figure 4.5: A flowchart describing the current-ramping algorithm.

With the weak links selected, and the current-ramping process finalised, we are now prepared to investigate how each of the weak-link approaches fare, in terms of both the accuracy of the power-law exponents they produce, and the time taken to run. The results from this analysis are presented and discussed in the next chapter.

Chapter 5

Results and Discussion

Various properties of the 'All-to-All' (AtA), 'Constant Radius Reduction' (CRR) and 'Constant p_r' (CPR) models will be presented, compared and discussed over the following sections.

The first three sections will discuss geometric properties of the three models. Section 5.1 will discuss the sizes of the groups produced by each of the models in turn. Section 5.2 will develop this further and explore the progression of the second moment for each model. Section 5.3 will then discuss the distributions of weak-link distances and neck width sizes collected by each of the three models.

The next two sections will focus on the transport properties of the three models. Firstly, section 5.4 will discuss the conductivities produced by the three models for various system sizes. Secondly, section 5.5 will look at the results obtained from the current ramp: both the voltage-current data, and the evolution of the critical current I_c with coverage.

Finally, the model performance will be discussed in section 5.6, and some final comments on the models will be made in section 5.7. All graphs used in the results section will follow the same colour-scheme when representing the coverages. Low coverages near $p = 0.68$ will be shown in blue, while high coverages up to $p = 0.8$ will be shown in red. The intermediate coverages between these two extremes will by represented by an appropriate colour in between blue and red. The exact colour scheme is shown in figure 5.1.

5.1 Group sizes

The first system property to be investigated is the set of group sizes within the system, for the original network and for the CRR and CPR models. The distribution of group sizes provides a first quantitative look into the system. The second moment is to be investigated in section 5.2 and is calculated directly from the set of group sizes present within each of the systems, so how the distribution of group sizes develops as the system coverage increases can provide insight into the second moment values observed at each coverage.

Figure 5.1: The colors, selected from a palette resembling the visible light spectrum, that are attributed to each coverage p in each of the graphs produced in the following sections. $p = 0.68$ is shown in deep blue. $p = 0.8$ is shown in deep red. Coverages between these two values are presented by colours selected from intermediate positions in the spectrum.

Figure 5.2 presents the group sizes associated with the percolating systems prior to the radius reduction and regrouping discussed in section 4. The group sizes are graphed against the frequencies with which they appear on the logarithmic scale for clarity. The group sizes were collected over the course of 100 trials of a 200x200 system. For each trial, the set of coverages to be examined between $p = 0.68$ and $p = 0.8$, at intervals of 0.005, were generated by iteratively adding clusters randomly to the preceding coverage until the desired coverage was produced. The spanning groups are included, and present themselves at the highest size values for each coverage. The significant feature is that, aside from the spanning groups located at group sizes greater than 10^4, the number of groups found at each size decreases as the coverage increases. The spanning groups, on the other hand, increase in size significantly - the increase in size from $10^{4.7}$ at $p \approx 0.72$ to $10^{4.9}$ at $p = 0.8$ represents a change of 50,000 clusters to 80,000 clusters in the size of the largest, spanning groups.

This is due to the fact that, as p increases, the gaps between various groups are bridged by newly deposited clusters. Larger groups have a larger boundary, and so have more places where a newly-deposited cluster could connect the group to the largest spanning group. This can be seen in the dramatic decrease in the number

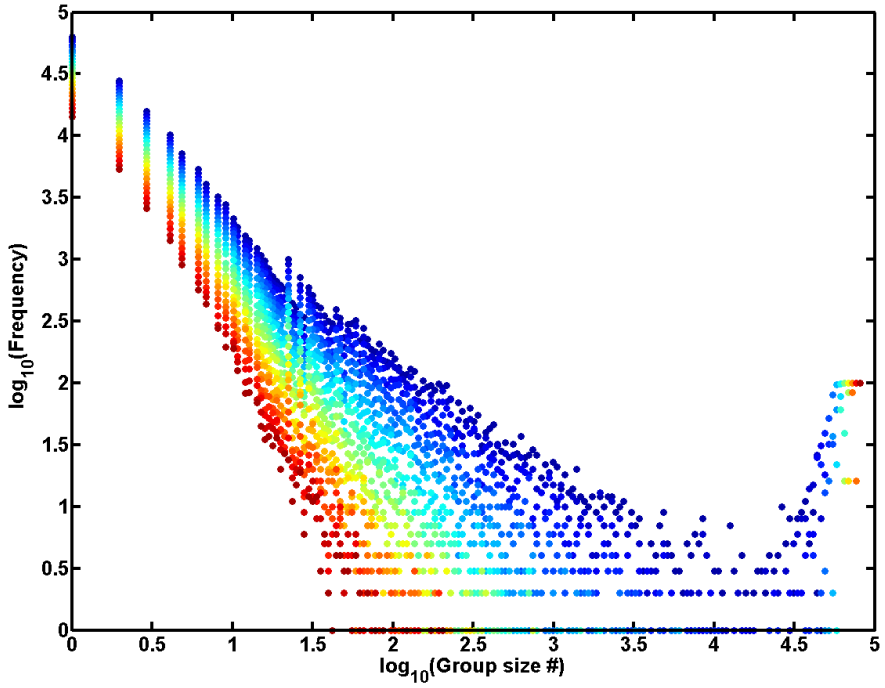


Figure 5.2: The frequencies at which various group sizes appear before radius-reduction and regrouping has occured, in a 200x200 system. The data is collected for coverages from $p = 0.68$ in blue to $p = 0.8$ in red in steps of 0.005, for each of the 100 trialled systems.

of groups found to have sizes between 10^2 and 10^3 at the coverage increases. In short, the number of all group sizes observed decreases as clusters recently added to the system connect otherwise-disconnected groups to the backbone, and the small groups are all slowly subsumed into the spanning group. With this in mind, it can be seen that if the large spanning clusters seen at group sizes above 10^4 clusters are ignored (as they will be when calculating the second moment), the average remaining group size appears to decrease as the coverage increases. This can be observed as the median value of the group sizes shifting left as the coverage increases.

Figure 5.3 presents the group sizes that remain following the radius reduction process associated with the CRR model, as discussed in section 4.0.4. The radius reduction process serves to break groups up into smaller fragments. The figure shows that as the coverage increases, the number of small groups reduces. However, the number of larger groups with sizes greater than 10^2 clusters increases as the coverage increases. This is due to the fact that the CRR model makes use of a constant radius reduction factor to produce the observed fragmentation. As the coverage increases and more clusters are deposited, connections are made between neighbouring groups. Some of these connections will also constitute weak links within the CRR model, but many of the new connections will not satisfy the weak-link criteria, and so will permanently join the two groups together. Because of this, the size of the largest groups will continue to increase as they are connected to more and more of the smaller groups by strong connections. A corrolary of this is that, as the number of small groups decreases and the number of larger groups increases, the average group size will get larger as the coverage

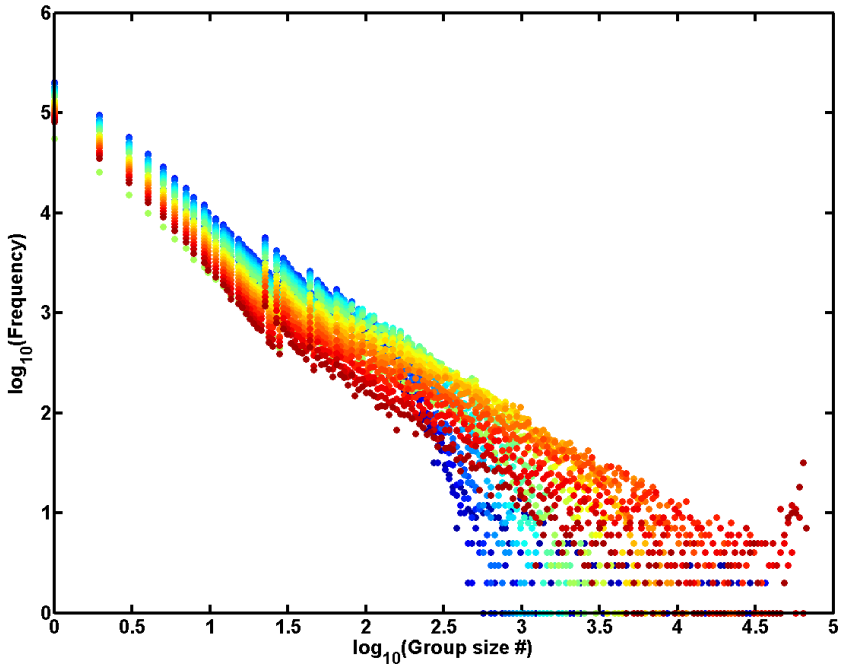


Figure 5.3: The frequencies at which various group sizes appear for the CRR model, in a 200x200 system. The data is collected for coverages from $p = 0.68$ in blue to $p = 0.8$ in red in steps of 0.005, for a total of 100 systems.

increases. This can be observed as the median value of the groups sizes shifting right as the coverage increases.

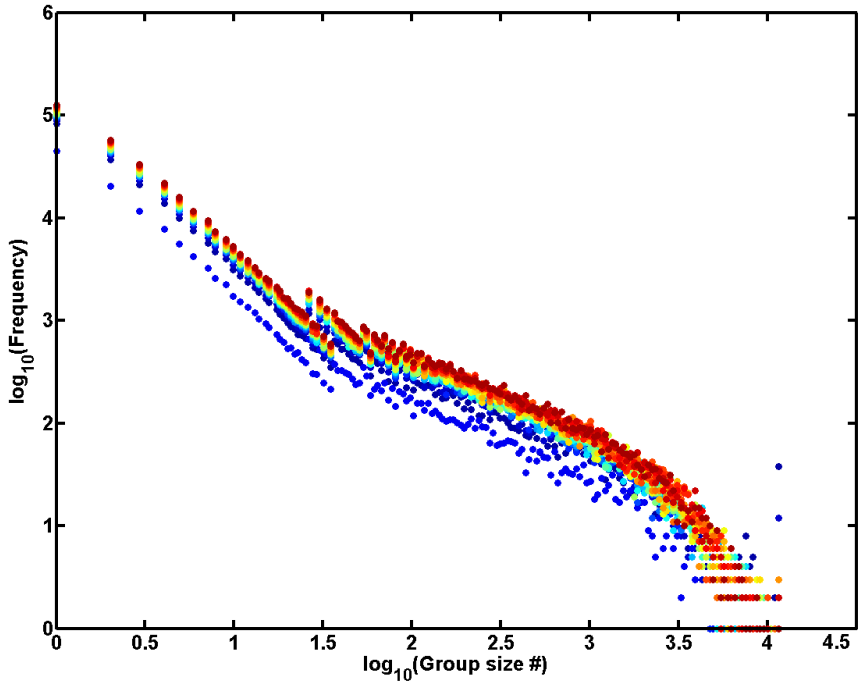


Figure 5.4: The frequencies at which various group sizes appear for the CPR model, in a 200x200 system. The data is collected for coverages from $p = 0.68$ in blue to $p = 0.8$ in red in steps of 0.005, for a total of 100 systems.

Figure 5.4 presents the group sizes that remain following the radius reduction process associated with the CPR model, as discussed in section 4.0.4. In the CPR model, the radius reduction process serves to break groups up into smaller fragments, but the radius reduction factor λ used to achieve this changes as the system coverage p changes, as seen in equation 4.7. An interesting observation to make about the group sizes associated with the CPR model is that the distribution of group sizes is almost exactly the same regardless of coverage, with only a frequency offset separating the group-size distributions demonstrated by each coverage. This makes sense if the distribution of groups is dependant on the coverage of the system where the clusters are grouped. In the original system, the groups are produced for each coverage p of the system. For the CPR model, however, the coverage of the system is reduced to a constant 'reduced coverage' p_r, using a radius reduction factor λ calculated in equation 4.7. This forces the system to group particles together as if the network was always at the same coverage of p_r, which would explain why the distribution of group sizes does not change when the coverage changes. Only the number of groups observed to have each group size changes, increasing with coverage. This is simply due to the fact the number of clusters in the system increases as the coverage p increases, so that there are more clusters available to form groups in the higher-coverage systems when the system is reduced to the constant reduced coverage p_r. Because the distribution of group sizes does not change as the coverage p changes, we can see that the median value does not shift either. This implies that the second moment will have the same value for all values of p when it is calculated for the CPR model.

The changes in the group sizes, observed in the group size figures, provide a clue as to how the second moment of the various models is expected to change when the coverage is increased. This will be observed and discussed with reference to the graphs of the second moments in the next section.

5.2 The second moment S

5.2.1 The original groups

The second moments for the original system, prior to radius reduction and re-grouping,are shown in figure 5.5 for three system sizes L: L =100x100, 200x200 and 400x400. The second moment S is calculated by finding the average group size of the original groups before the radius reduction factor from chapter 4 is introduced and any regrouping occurs. The main purpose is to confirm that the power-law relationships predicted by percolation theory are observed in the simulated systems. The finite-size effects discussed in section 2.2.2 can influence network statistics at coverages near p_c. As S is purely a geometric measure, given by the average group size in the system, S can be used as evidence to confirm the exact coverages where the network statistics begin to go awry.

The second moments of both the 100×100 and 200×200 systems are averaged over 100 trials, whereas the 400×400 system is averaged over 15 trials, leading

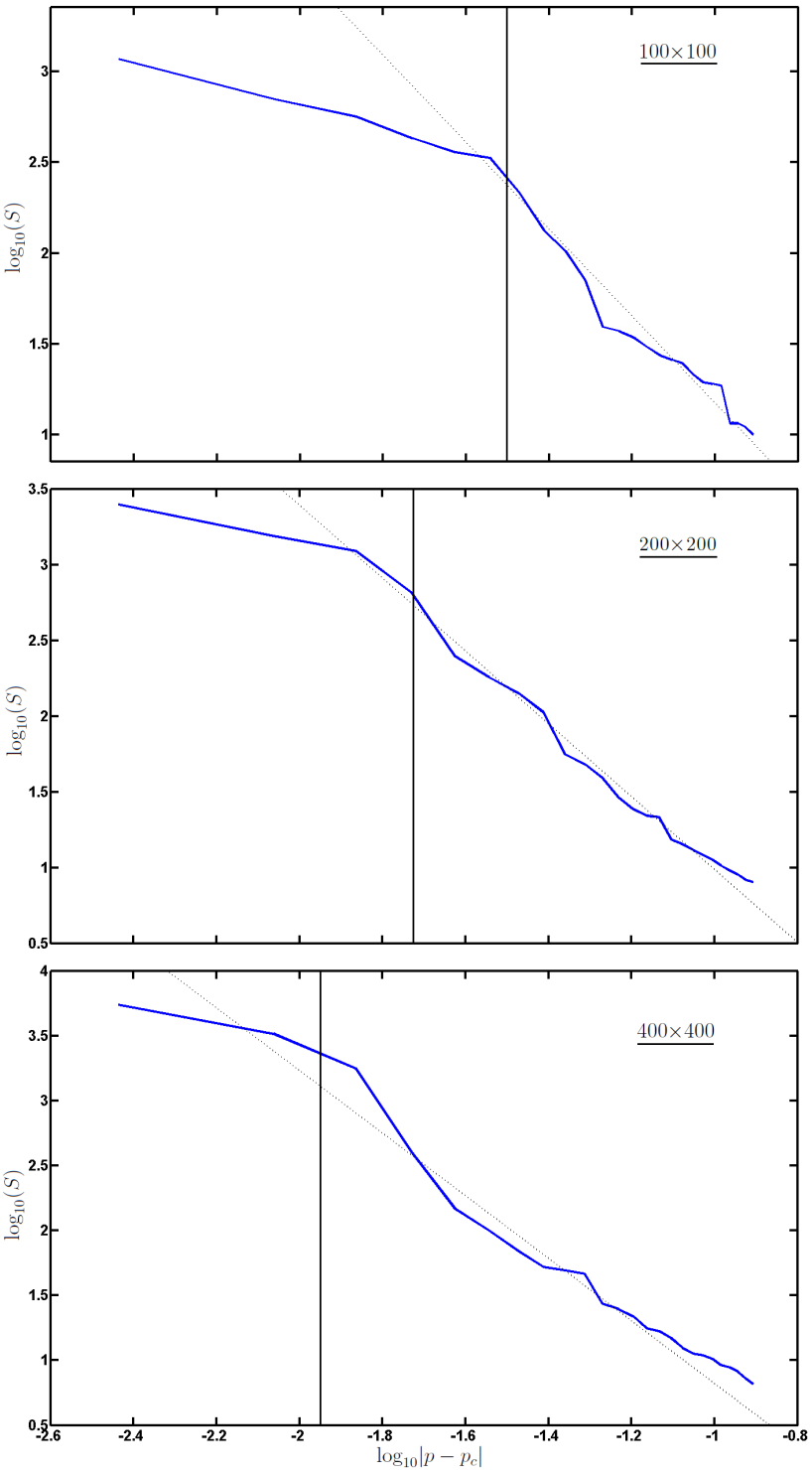


Figure 5.5: The second moments of original 100x100, 200x200 and 400x400 systems as a function of $|p - p_c|$. A line with slope $-\gamma$ has been added for comparison, where $\gamma = \frac{43}{18}$. Both the 100x100 and 200x200 systems are averaged over 100 trials, while the 400x400 system is averaged over 15. The solid vertical line in each plot shows the value of $\log_{10}|p - p_c|$ below which the second moment S is expected to deviate from the slope of $-\gamma$ due to finite size effects.

to more noticable fluctuations. Recall that S is given by equation 2.1, where $S \propto (p - p_c)^{-\gamma}$, where $\gamma = \frac{43}{18}$. A line with slope $-\gamma$ has been added to each of the three plots for reference. The data for all three system sizes is well aligned with the expected slope aside from small fluctuations about the expected slope, and a deviation from the expected slope at low coverages which is worth noting in all three system sizes. The close alignment of S and the expected slope reinforces the claim that equation 2.1 appropriately describes the relationship between the second moment and the coverage p.

It is possible to calculate where the system statistics like the second moment should begin to deviate from the expected power-law by applying the finite size scaling approach mentioned in section 2.2.2. In this context, we expect the network to stop obeying the usual percolation power-laws when the percolation correlation length ξ_p exceeds the system size, so that $|p - p_c|L^{\frac{1}{\nu}} < 1$, and thus $|p - p_c| < L^{\frac{-1}{\nu}}$ [8]. To apply this to figure 5.5, we will instead compare $\log_{10}|p - p_c| < \frac{-\log_{10}(L)}{\nu}$.

For the 100x100 system, this suggests that the second moment should deviate from its expected value at $\log_{10}|p - p_c| < \frac{-\log_{10}(L)}{\nu} = \frac{-3\log_{10}(100)}{4} = -1.5$. This corresponds to a system coverage p of 0.708 and corresponds with the dramatic shift in the slope of the second moment data as the coverage tends towards p_c from above.

For the 200x200 system, the second moment is expected to deviate from its expected value at $\log_{10}|p - p_c| < \frac{-3\log_{10}(200)}{4} = -1.726$, corresponding to $p = 0.695$. Again, this corresponds to the change in slope of figure 5.5(b).

For the 400x400 system, $\log_{10}|p - p_c| < \frac{-3\log_{10}(400)}{4} = -1.952$, corresponding to $p = 0.688$. The 400x400 data is more difficult to compare with the value produced by the finite-size analysis, due to the smaller number of trials and the larger impact of fluctuations. Nevertheless, this value for $\log_{10}|p - p_c|$ again appears to coincide with the change in slope of the second moment data. The fact that these values are predicted by finite size scaling theory suggests that the low $\log_{10}|p - p_c|$ values of the system are unreliable for obtaining accurate power-law exponent. As finite size effects dominate at small values of $|p - p_c|$, we must use data from high coverages to calculate γ.

It is worth noting that at high coverages the number of groups that are separate from the infinite, spanning group become few and far between. With so few groups, the second moment is calculated using fewer sample groups and so the second moment becomes more open to being skewed by outliers. This leads to greater fluctuations in the value of the second moment as p increases. This is another product of the system size, as a larger system provides more space for additional groups to exist at the same coverage, and so it should be expected that there are larger fluctuations in the second moment as the coverage increases towards 1. Nevertheless, it appears that there is a large range of $|p - p_c|$ values left unaffected by the finite size of the systems being investigated, in which the correct value of $\gamma = \frac{43}{18}$ is obtained.

5.2.2 The 'Constant p_r' and 'Constant Radius Reduction' models

CPR

As discussed in section 4.0.4, the foundational idea of the constant p_r (CPR) model is that the radii of the deposited clusters are reduced so that the coverage of the sample post-reduction is a constant coverage value p_r. Clusters that still touch each other continue to form groups.

In principle, because these groups are still formed from a network of equally sized discs located at random points in the system, they should still obey the usual second moment power law discussed above in the AtA case. The significant difference here is that, as far as the groups are concerned, the coverage is always equal to a single value - p_r. The second moment in the CPR model should remain constant, because the grouping is performed on the system as if $p = p_r$, regardless of the pre-reduction value of p. This means that the value of $|p - p_c|$ will remain constant, as $|p - p_c| = |p_r - p_c|$ following the reduction of the cluster radii, and so S will remain constant, since $S \propto |p - p_c|^{-\gamma}$. Figure 5.6 presents the second moment of the CPR system as a function of pre-reduction coverage. Aside from the initial two points at $\log_{10} |p - p_c| < -1.9$, which result from the same finite size effects found in the raw AtA system, the second moment is found to remain roughly constant, as expected. The conclusion is that, in the CPR model, the average size of groups stays roughly constant post-reduction. Reducing the value of p_r would reduce the apparent coverage of the system. This would increase the size of $|p - p_c|$, and so would decrease the second moment as given by equation 2.1.

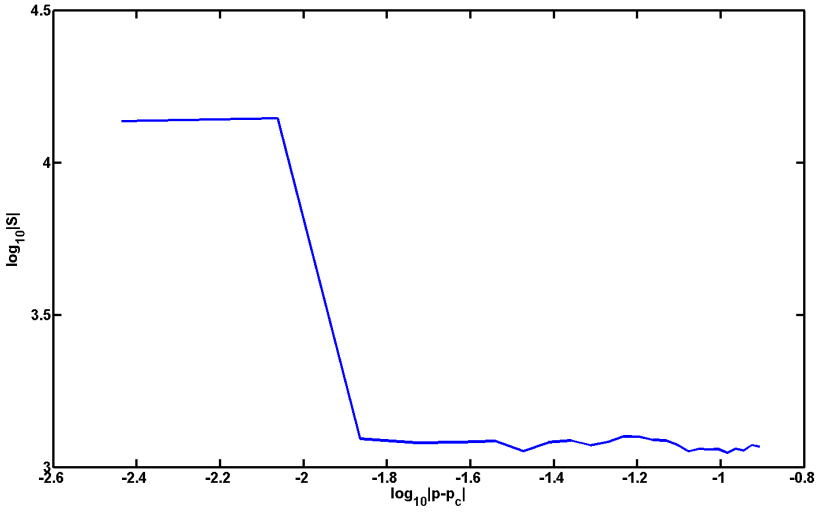


Figure 5.6: The second moment of 400x400 CPR systems as a function of $|p - p_c|$. The data was averaged over 100 trials.

CRR

As discussed in section 4.0.4, the constant radius reduction (CRR) model is produced by reducing the radii of the deposited clusters by a constant proportion λ, called the radius reduction factor. Clusters that still touch each other continue to form groups. The data presented in the following sections has been produced using $\lambda = 0.85$.

The second moment for the CRR model, shown in figure 5.7, possesses significant curvature. It seems as if the second moment begins to increase asymptotically as the coverage approaches $p \approx 0.8$. This is due to the fact that the CRR network of resistors is generated by reducing the radius of every particle by applying the radius reduction factor λ, and finding the previously-existent connections that are broken by the process as is mentioned in section 4.0.4. The apparent coverage of the system is p_r, and is found using equation 4.5.

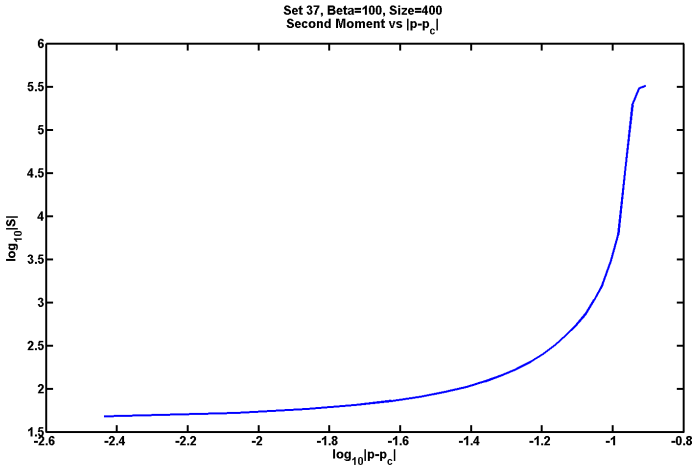


Figure 5.7: The second moment of 400x400 CRR systems as a function of $|p - p_c|$. The data was averaged over 100 trials.

There must be some coverage p^* for the normal system that, when the cluster radii of this p^* system are reduced by λ, the system's reduced coverage remains at p_c and at least one group continues to span the system. We can calculate the original coverage that causes this by applying equation 4.5 where $p_r(p, \lambda) = p_c$:

$$\begin{aligned} p_c &= 1 - (1 - p^*)^{\lambda^2} \\ 1 - p_c &= (1 - p^*)^{\lambda^2} \\ (1 - p_c)^{\lambda^{-2}} &= 1 - p^* \\ p^* &= 1 - (1 - p_c)^{\lambda^{-2}} \end{aligned} \tag{5.1}$$

As λ tends towards 0, the value of p^* will tend towards 1, and as λ tends towards 1, p^* will tend towards p_c. At $\lambda = 0.85$ and $p_c = 0.676339$, the coverage p^* at which we expect 50% of the networks to continue spanning is found by equation 5.1 as being equal to

$$p^* = 1 - (1 - 0.676339)^{0.85^{-2}} = 0.7901$$

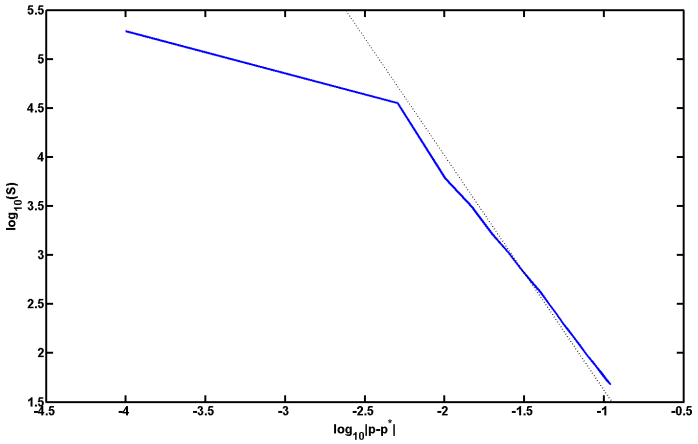


Figure 5.8: The second moment of 400x400 CRR systems, related to $|p-p^*|$ rather than $|p-p_c|$. A line with slope $-\gamma$ has been added for comparison, where $\gamma = \frac{43}{18}$. The data was averaged over 100 trials.

This coverage of $p^* = 0.79$ corresponds with $\log_{10}(p - p_c) = -0.944$. Now, for the reduced network created by the CRR model, the pre-reduction coverage p^* is reduced to p_c. The critical coverage p_c is defined in section 2.1 as being the coverage at which 50% of systems contain a spanning group. For the CRR model, however, the critical coverage at which this occurs is actually given by p^*. With that in mind, it is worth investigating whether we recover the usual second moment power-law exponent when we use $\log_{10}|p - p^*|$ rather than $\log_{10}|p - p_c|$. Figure 5.8 presents the same S data as figure 5.7, but with the coverage values compared with p^* rather than p_c. The data is compared with the expected slope of $-\gamma = \frac{-43}{18}$, and reaches reasonable agreement with this expected slope. The CRR model fits the expected second moment power law when it is treated as approaching p^* from below, because the reduced coverage p_r is less than p_c for all $p < p_c$. As $p \to p^*$, the sizes of the largest groups begin to diverge, and at p^*, can be expected to start spanning the system.

5.3 Distance and neck width distributions

The distance between connected clusters determines the width of the neck joining them, and so determines the conductivity between the two clusters. Investigating which distances between clusters are more common will be a valuable task, as it will help uncover the distribution of conductances in the system and will launch the exploration of the system conductivity.

5.3.1 The 'All-to-All' distribution

The distribution of cluster separations for the current system is shown in figure 5.9 at coverages from $p = 0.68$ to $p = 0.8$ with spacings between coverages of 0.005. Both the un-normalised and the normalised data are present. In the un-

normalised plot, the number of connections present at every cluster separation distance d increases as the coverage increases from $p = 0.68$ to $p = 0.8$. The normalisation of the second graph is performed on each coverage by simply dividing the number of connections found with a particular separation by the total number of connections found at that coverage. Within the normalised data, it is possible to see that the peak in the frequency occurs near $d = 0.8$ at $p = 0.68$, shifting left to a position around $d = 0.65$ at $p = 0.8$ as the coverage increases.

For comparison with the CRR and CPR data, it is worth remembering that certain connections are discarded as potential weak links based on whether the distance between the clusters forming the connection is greater than the threshold value d_0, as constructed by equation 4.4. These threshold values have been marked on figure 5.9. If the distance between cluster centers is greater than d_0, then that connection is a potential resistor and so is kept. If the distance between them is less than d_0, then the connection is considered superconducting and so is discarded. In the CRR model, for a chosen radius reduction value of $\lambda = 0.85$ and cluster radius $r = 0.5$, the threshold distance is simply $d_0 = 0.85$. This CRR threshold is indicated on figure 5.9 as a blue dotted line. In the CPR model, the radius reduction value depends on p and the chosen reduced coverage p_r, as shown in equation 4.7, and establishes a minimum distance threshold for p, again according to equation 4.4. For $p_r = 0.65$, the radius reduction values $\lambda(p)$ for the lowest and highest coverages are $\lambda(0.68) = 0.96$ and $\lambda(0.8) = 0.81$ respectively. These extremal values are indicated on figure 5.9 as two red dashed lines. The threshold values associated with coverages between $p = 0.68$ and $p = 0.8$ are located in the intervening space, which is coloured pale red.

The significance of figure 5.9 is that the distribution of distances between clusters, and the evolution of this distribution with increasing coverage, affects the average conductivity of each connection and the proportion of connections that satisfies threshold conditions.The connections between clusters are given conductivities proportional to the neck widths separating them, so that $g \propto w$. The above distance data has been transformed into the equivalent neck widths using equation 4.1, where

$$w = \sqrt{(2r)^2 - d^2} = \sqrt{1 - d^2}$$

The distribution of neck widths is shown on figure 5.10. The CRR and CPR thresholds have also been transformed and displayed on this figure, so that they represent the maximum neck width allowed. Again, in the CPR model, this maximum neck width threshold depends on the coverage. The CRR neck width threshold w_r is found at $w_r = \sqrt{1 - 0.85^2} = 0.529$ as a blue dotted line. The CPR threshold are given by $w_p(\lambda) = \sqrt{1 - \lambda^2}$. The minimum-coverage CPR threshold $w_p(0.68)$ is found to be $w_p(0.68) = \sqrt{1 - 0.96^2} = 0.280$, whereas the maximum-coverage CPR threshold is $w_p(0.8) = \sqrt{1 - 0.81^2} = 0.580$. Red dashed lines are positioned at these neck widths to indicate the threshold values, and the thresholds for the intervening coverages are located in the pale pink area.

The purpose of introducing the CRR and CPR thresholds here is to estimate how the choice of threshold might affect the distribution of weak-link conductances present in the system. Looking at the unnormalised data for the 'All-to-All' model in figure 5.10, it is easy to see that the number of weak links increases as the

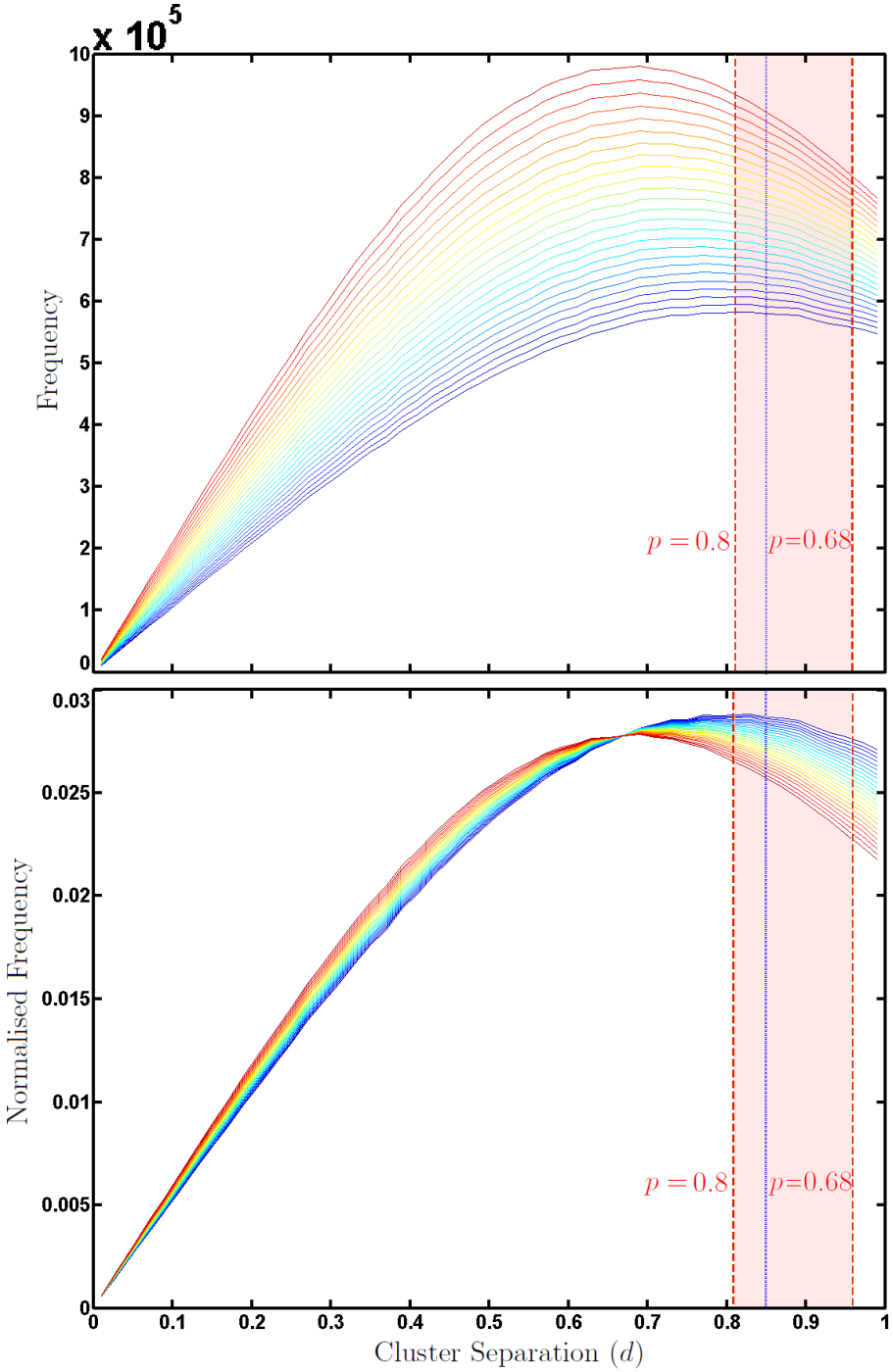


Figure 5.9: The frequencies of cluster separations, plotted for each coverage p. Both the unnormalised data, and the same data normalised by the total number of links in the system, are present. The minimum separation threshold allowed by the CRR and CPR approximations is also shown. The minimum separation allowed by CRR is indicated by the dotted blue line, located at $d = 0.85$. The minimum distances allowed by CPR vary with the coverage, and so only the thresholds associated with $p = 0.68$ and $p = 0.8$ are explicitly shown as red dashed lines, located at $d = 0.96$ and $d = 0.81$ respectively. The thresholds for the intermediary coverages lie within the pale red area.

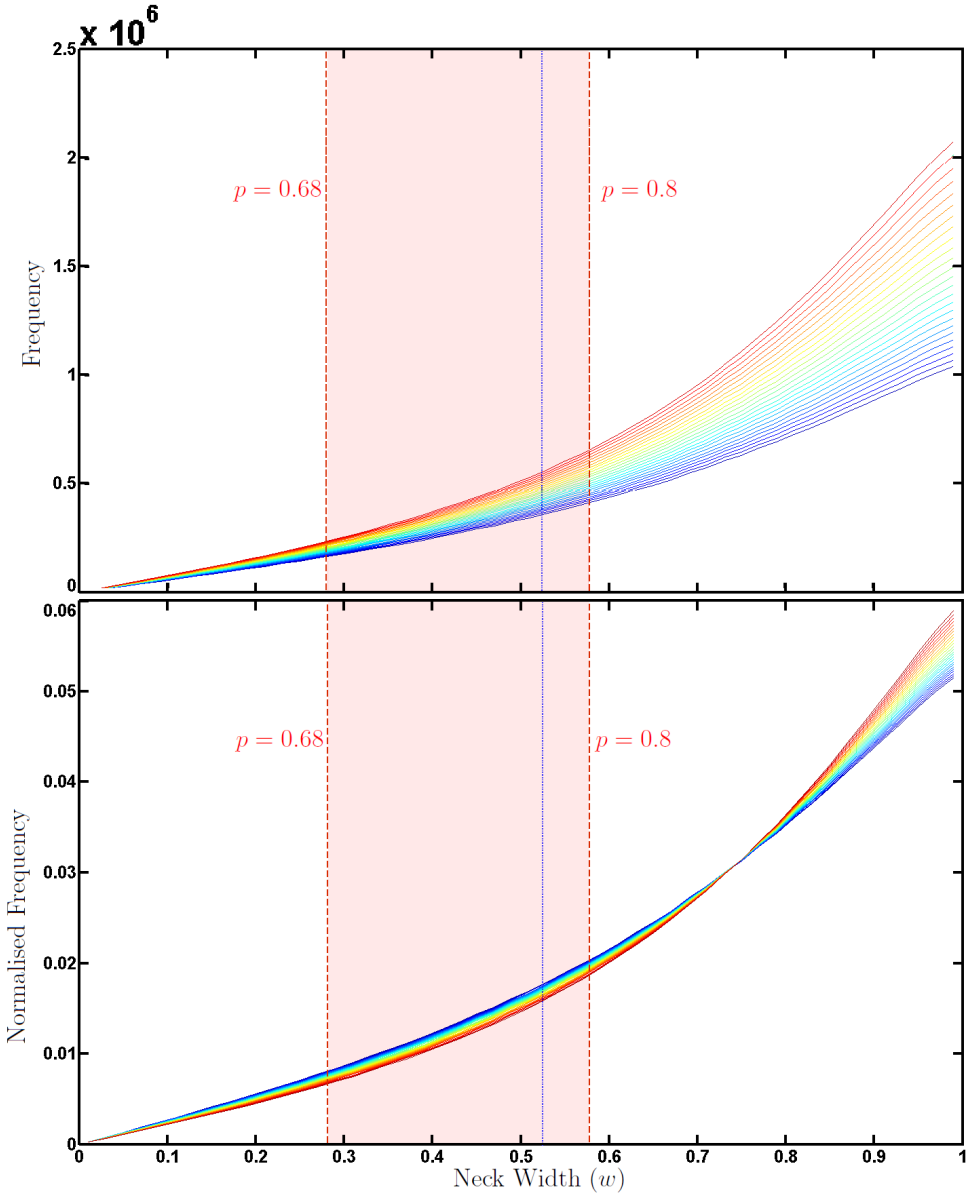


Figure 5.10: The frequencies of neck widths, plotted for each coverage p. Both the unnormalised data and the same data normalised by the total number of links in the system are present. The maximum neck width threshold allowed by the CRR and CPR approximations is also shown. The maximum neck width size allowed by CRR is indicated by the dotted blue line, located at $w_r = 0.529$. The maximum neck width sizes allowed by CPR vary with the coverage, and so only the thresholds associated with $p = 0.68$ and $p = 0.8$ are explicitly shown as red dashed lines, located at $w_{p-} = 0.280$ and $w_{p+} = 0.580$ respectively. The thresholds for the intermediary coverages lie within the pale red area.

coverage increases. This suggests that the system's conductivity should progress with an increase in coverage like a regular random resistor network if every weak-link is in its resistive state. If so, the usual value of 1.3 for the conductivity exponent t (from equation 2.3) should be recovered when the conductivity is calculated for multiple coverages.

Interestingly, the probability distribution for the neck widths does not remain the same as the coverage increases. As can be seen in the un-normalised plot, the proportion of neck widths where $w < 0.7$ decreases as the coverage increases, while the proportion of neck widths where $w > 0.8$ increases with coverage. There is a region in between $0.7 < w < 0.8$ where a cross-over between these two cases occurs. The threshold value associated with the CRR model is at $w = 0.529$. This is well below $w = 0.7$, and so sits decidedly in the region where the proportion of all neck widths contained within the threshold decreases as the coverage increases.

This suggests that the proportion of all links that qualify as weak links falls as the coverage increases, when only the threshold is considered in the CRR model. We consider the proportion of links that remain rather than the total number of links because, in the CRR and CPR models, links only qualify as weak links if their removal splits a group into multiple fragments, as discussed in section 4.0.3. This suggests that if the proportion of all links that meet the threshold criteria is large, then there is a good chance that they will be situated on the edges of blobs and so are likely to split groups up, resulting in a large number of weak links. As the proportion of all links that meet the CRR threshold decreases as the coverage increases, the expectation is that the total number of resistors constructed by the CRR method will also decrease as the coverage increases. If so, CRR systems would have more in common with random superconducting networks than random resistor networks, and so they should not be expected to follow the usual RRN conductivity power-law, given by equation 2.3.

5.3.2 The CRR and CPR data

A significant point of difference between the AtA model and the CRR and CPR models is that the two λ-based models remove a large number of potential weak links from consideration. A comparison of the network of weak links produced by the three different models is presented in figure 5.11 at two different coverages of $p = 0.69$ and $p = 0.75$. Weak links have been coloured green, and the remaining 'strong' connections of the CRR and CPR models have been coloured black. The AtA model considers every connection to be a weak-link by definition, and so is entirely green. By comparison, both the CRR and CPR models feature a significant number of black, 'strong' connections scattered throughout the system. At low coverages, the CPR model has a high value of λ and so only identifies a small number of connections as weak links. The CRR model, on the other hand, features a reasonably large proportion of weak links. By $p = 0.75$, though, the two networks are hardly distinguishable. The point to emphasise is that the number of weak links featured in either of the two systems is significantly smaller than the number of weak links presented by the AtA system.

The data for connections that finally qualify as weak links under the CRR model, for both weak-link separation distances and neck widths, is provided in figure 5.12. Figure 5.12 validates this prediction, with the number of weak links dropping for all separations and neck widths as the coverage increases. The neck width graphs imply that not only does the largest group size diverge as p approaches $p^* = 0.79$

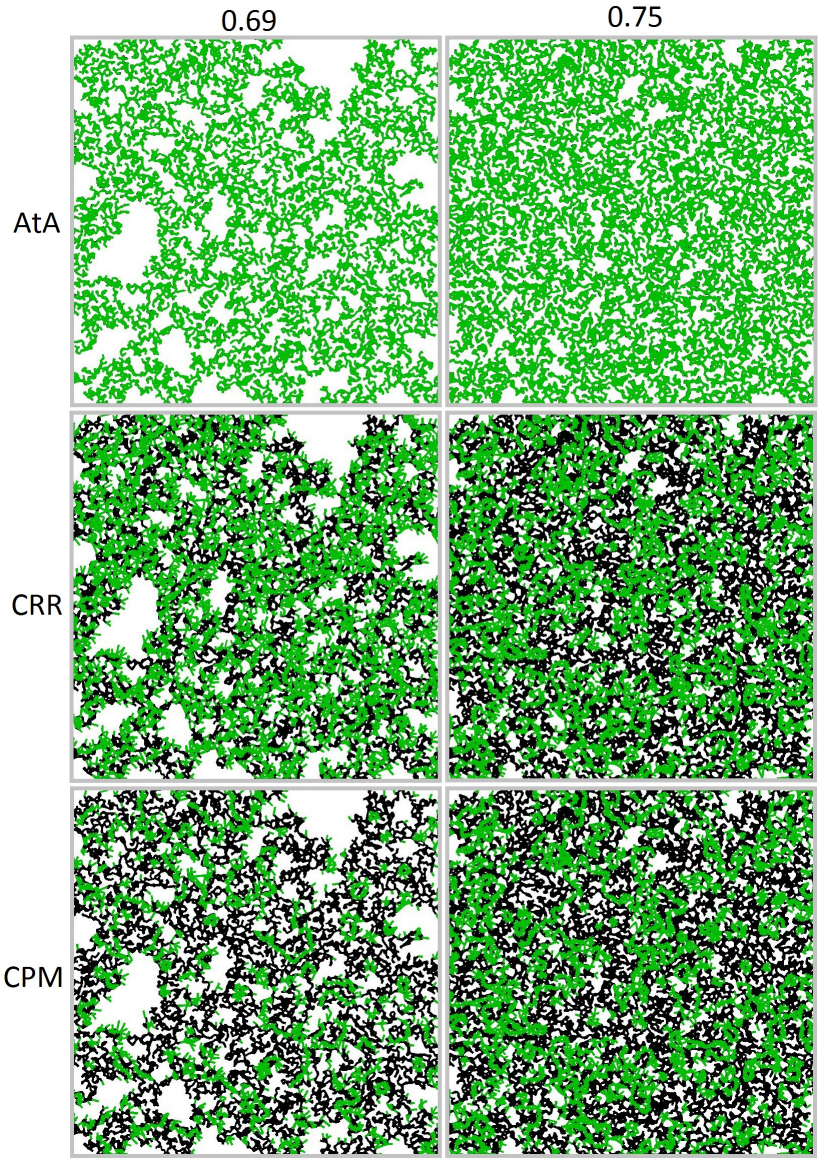


Figure 5.11: An illustration of how the three models - All-to-All (AtA), Constant radius reduction (CRR) and Constant p_r (CPR) - affect the selection of weak links in the network. The weak links are shown in green for the three models at two coverages, $p = 0.69$ and $p = 0.75$. The left-over 'strong' links are coloured black.

from below, as seen in section 5.2, but the number of weak links in the network decreases significantly leading up to $p = p^*$.

Similarly, the data for the connecitons that qualify as weak links under the CPR model is provided in figure 5.13. The number of connections found at each neck width increases as p increases. It is worth noting that the distributions are all linear at small neck widths and large separations, and that the only difference in distributions comes at neck widths near the threshold, where the recorded frequencies experience a noticable upturn. However, the upturn occurs at the same widths where similar increases in frequency are found in figure 5.10. This suggests that the curvature found in the CPR data is not due to the method itself, but instead reflects the same distribution as the AtA model. Any differences in

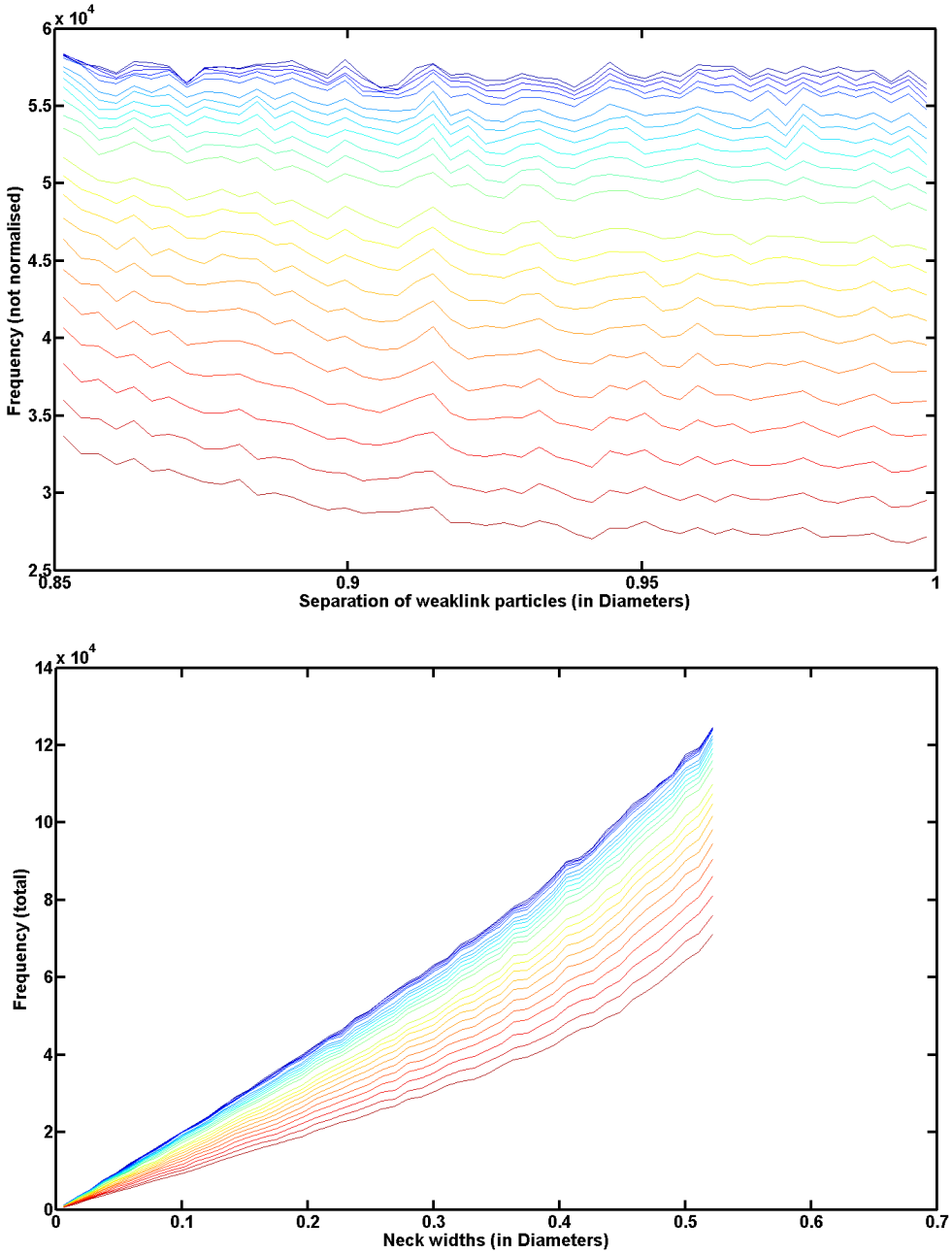


Figure 5.12: The top graph shows the frequencies of various separation distances for the weak-link connections produced by the CRR model. The bottom graph shows the frequencies of various neck widths for the CRR model. These are plotted for each coverage p in a 200x200 system.

conductivity between the AtA and CPR models should be due to the changing threshold of the CPR model rather than an inherent difference in the distributions of neck widths.

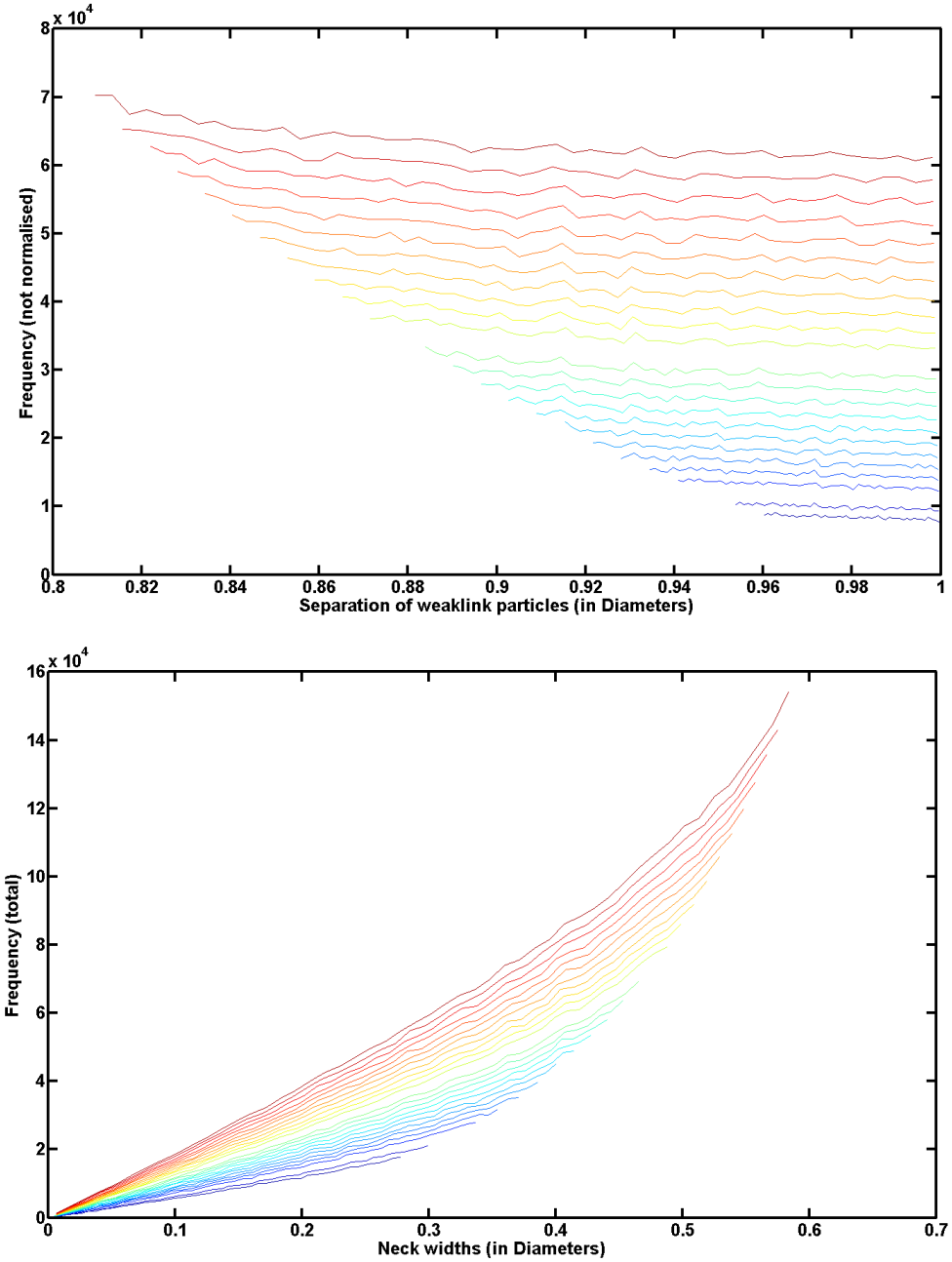


Figure 5.13: The top graph shows the frequencies of various separation distances for the weak-link connections produced by the CPR model. The bottom graph shows the frequencies of various neck widths for the CPR model. These are plotted for$p = 0.68$ in blue to $p = 0.8$ in red, at intervals of 0.005, in a 200x200 system. It should be noted that the extremal points of the frequency plot shift as the coverage increases, due to the change in weak-link threshold associated with the CPR model.

5.4 System conductivity

5.4.1 'All-to-All'

As discussed in sections 3.3 and 3.4, the normal-state conductivity of the system is calculated by converting the system into a network of resistors, then applying

Kirchoff's laws to determine the current flow through the system for a particular voltage difference across the sample. The conductivity can be retrieved from this by applying Ohm's law. The 'All-to-all' (AtA) model has every single connection participating as a weak-link in the network. In order to calculate the normal-state conductivity of the AtA network, each weak-link is treated as a resistor with a resistance proportional to its neck width. With this in mind, the conductivity of the AtA system is expected to depend on the coverage in a power-law fashion, with a form given by equation 2.3 with the exponent $t = 1.3$. The conductivity of the 100x100, 200x200 and 400x400 AtA systems, as a function of coverage, is presented in figure 5.14 with a line of slope 1.3 added for reference.

The slope of $t = 1.3$ fits the data well for each of the three system sizes over a wide range of $|p - p_c|$ values. However, finite-size effects appear to influence the data as p gets closer and closer to p_c: all three system sizes begin to deviate significantly from the expected slope at the same $\log(p - p_c)$ values as in figure 5.5. Otherwise, the conductivities that were calculated for the systems were extremely similar to the values predicted by standard percolation theory, and the slope of the modelled data follows the expected slope of $t = 1.3$ very well, with a very slight upturn of the slope at higher coverages in all three system sizes. The significant point is that the AtA approach is extremely similar to standard resistor-based continuum percolation models [8] in both system statistics (figure 5.5) and the evolution of conductivity with increasing coverage. The AtA model thus provides the expected function for exploring the normal states of the CRR and CPR models.

It is worth re-emphasising here that previous studies have all computed the conductivity of continuum systems by assigning a continuum-like distribution of conductances to the bonds of a lattice, rather than by calculating the conductivity from a full continuum deposition in the manner displayed here.

5.4.2 CRR and CPR

While the AtA model is certainly similar to the classical lattice random resistor network, the concept of incorporating weak-link resistors defined by their geometry in the continuum is an untouched subject, and so the usual power-law relationship expected of the AtA model should not necessarily be expected to reveal itself in the conductivities of the CPR and CRR models. Both the conductivity plot for the constant p_{min} system, shown in figure 5.15, and the conductivity plot for the constant radius reduction system, shown in figure 5.16, deviate from the expect power-law behaviour that is demonstrated by the 'All-to-All' model in figure 5.14. They both suffer from the same finite size effects at low p values that influenced the AtA model.

Figure 5.15 shows the CPR conductivity data assuming a slope of 1.65 at larger system sizes. This constant slope should define the conductivity exponent t for the system, yet the fact that it is larger than t's universal value of 1.3 implies that the CPR system demonstrates a progression of conductor distributions that is unusual enough to no longer replicate the inverse Swiss cheese model, a possibility mentioned in section 2.5.3. The investigation into the neck widths that qualified

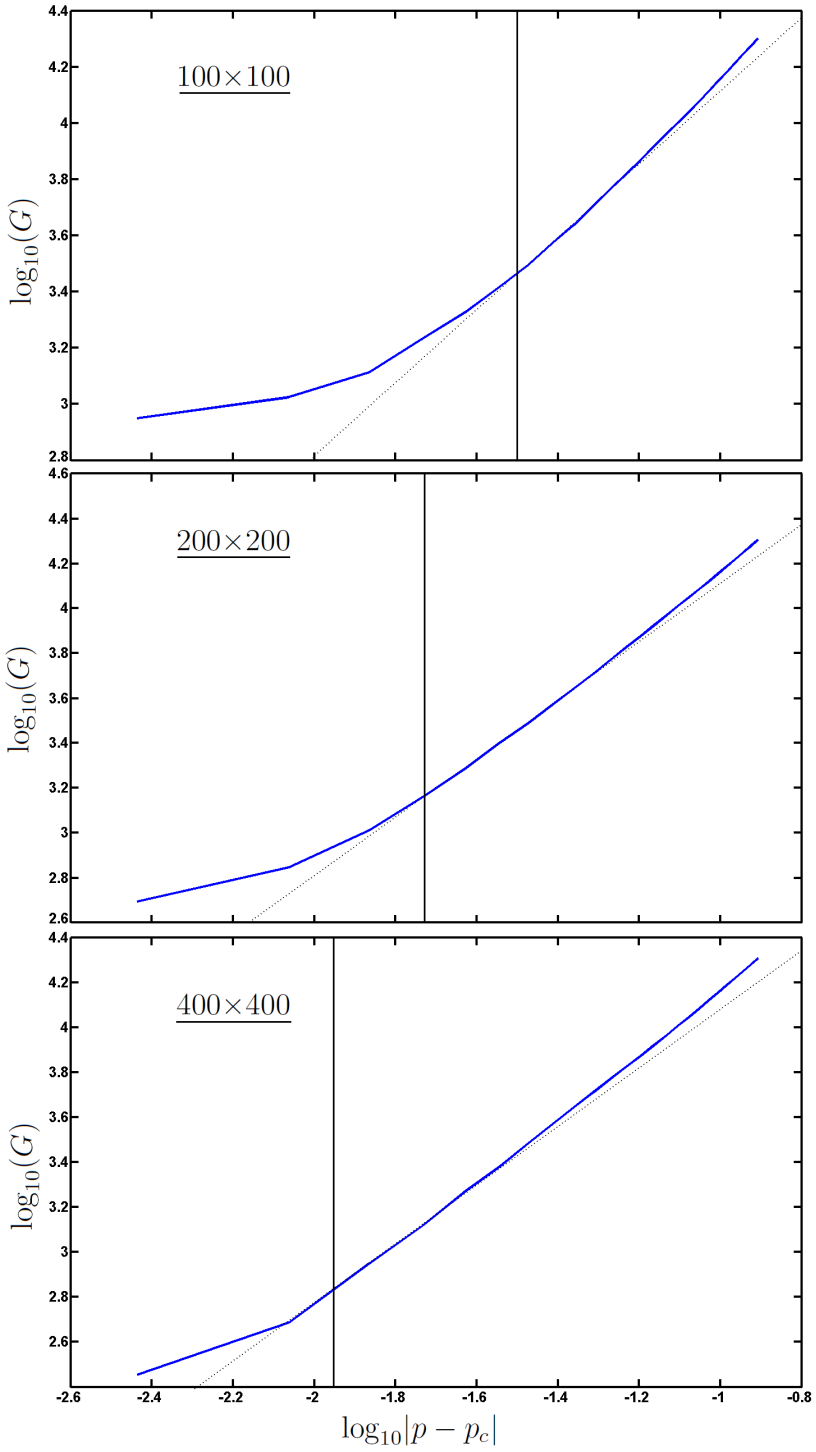


Figure 5.14: The conductivity of the 'All-to-All' network, graphed against the difference between the coverage p and p_c, for three system sizes 100x100, 200x200 and 400x400. Both the 100x100 and 200x200 systems are averaged over 100 trials, while the 400x400 system is averaged over 15. A line with a gradient of 1.3 has been added to each plot for reference.

for the CPR model suggests that this discrepancy is due to the changing threshold of connections, rather than some change in the distribution of neck widths which govern the conductivity of the connections. The increase in the maximum neck width threshold w_p, due to the increase in coverage, causes the difference in

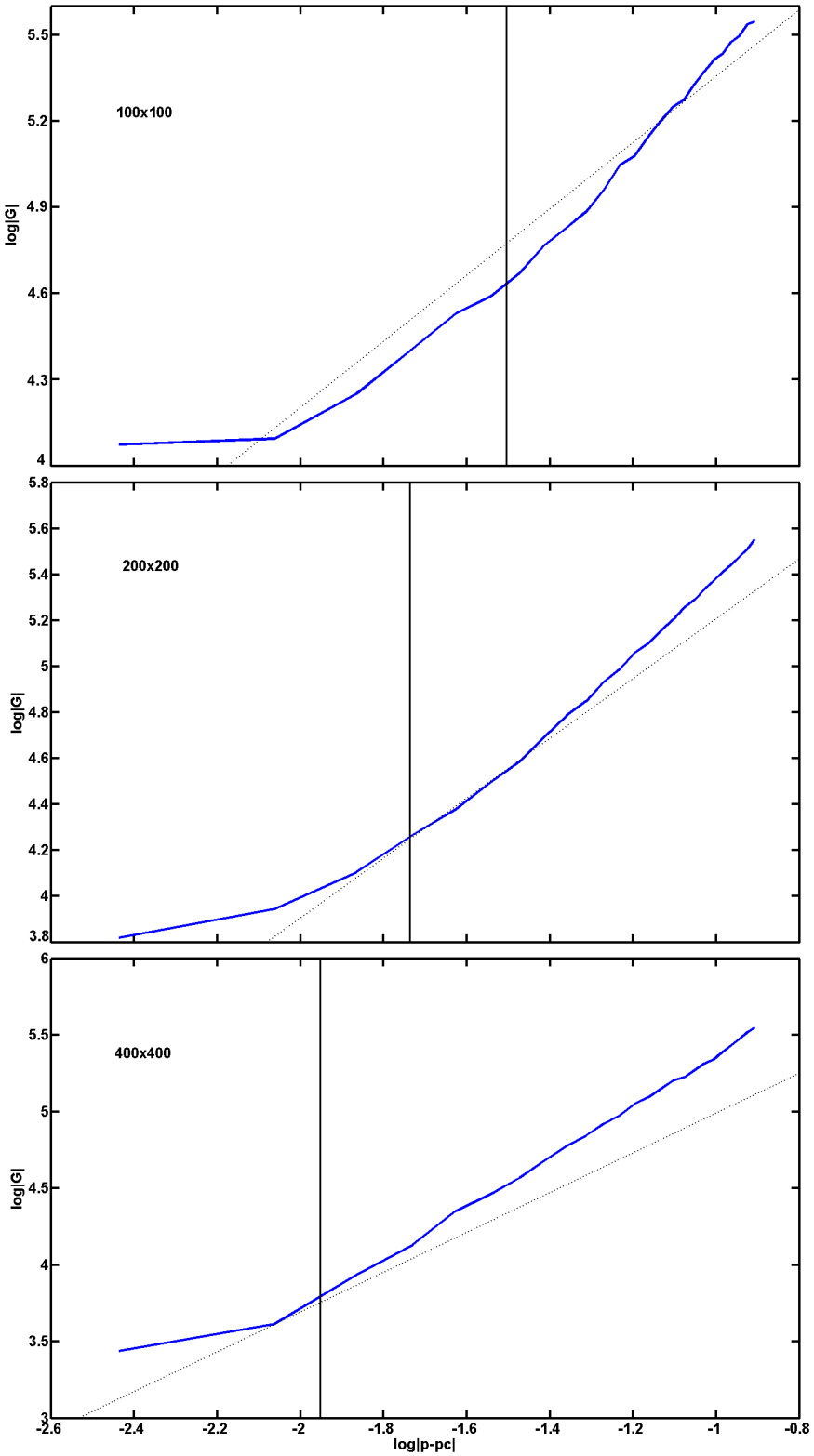


Figure 5.15: The conductivity of the 'constant p_r' network, graphed against the difference between the coverage p and p_c, for three system sizes 100x100, 200x200 and 400x400. All three systems are averaged over 100 trials. A line with a gradient of 1.3 has been added to each plot for reference.

the conductor distributions of the AtA and CPR models - while the distribution of neck widths stays reasonably constant in the AtA model (give or take a scaling

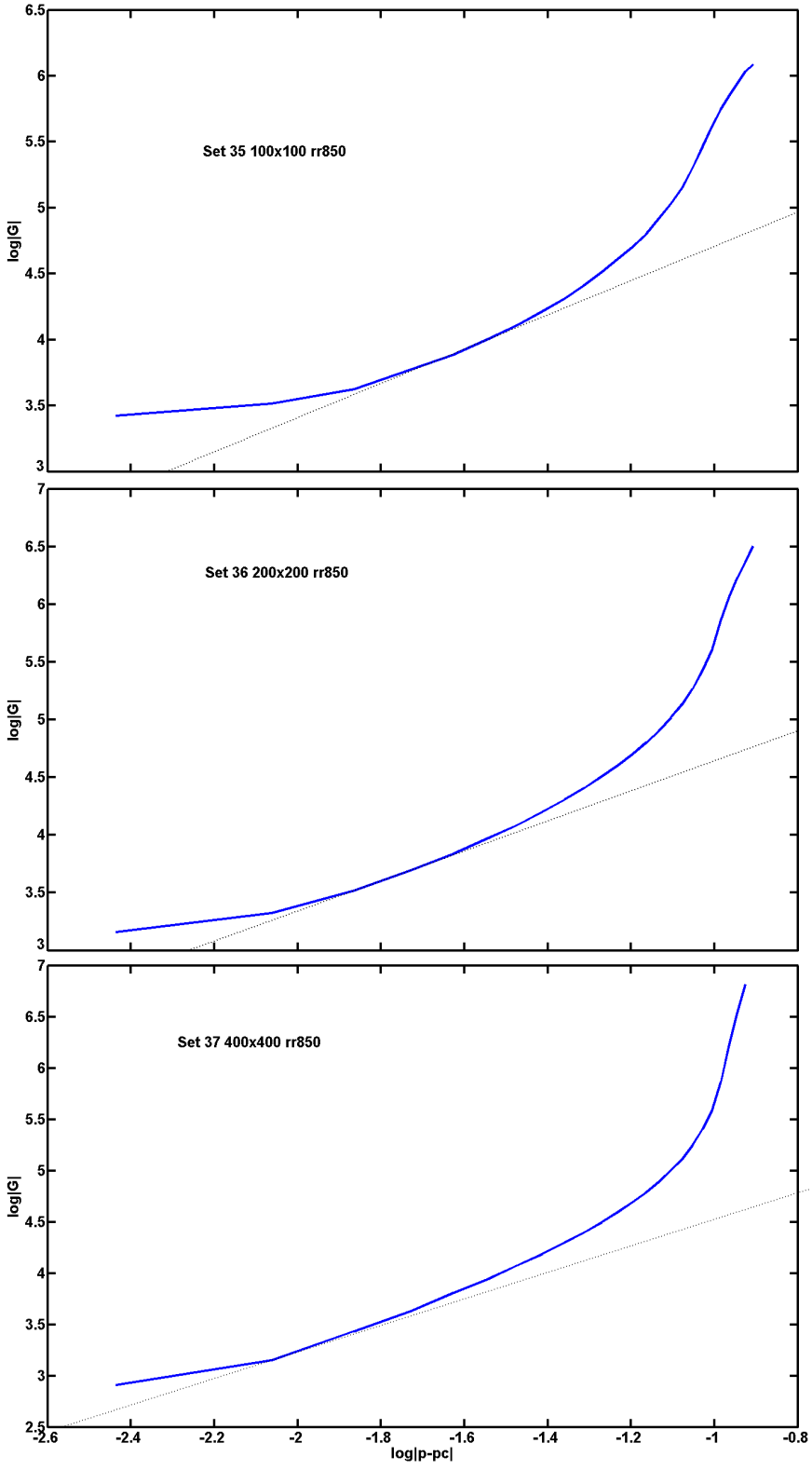


Figure 5.16: The conductivity of the 'constant radius reduction' network, graphed against the difference between the coverage p and p_c, for three system sizes 100x100, 200x200 and 400x400. All three systems are averaged over 100 trials. A line with a gradient of 1.3 has been added to each plot for reference.

factor), the changing upper threshold of the CPR model introduces an influx of new conductors at the highest neck width range as the coverage increases. The introduction of new, large neck widths artificially inflates the measured conductivity of the system as the coverage increases, beyond what would be expected by regular resistor networks and the AtA model.

The plotted data for the different system sizes in figure 5.16 have significant curvature on the log-log scale, implying that the conductivity rises significantly more rapidly as the coverage increases than the AtA and CPR models. This is a result of the systematic decrease in possible conductors seen in figures 5.10 and 5.12. Not only are weak links being short-circuited or covered up by newly deposited clusters as the coverage is increased, but the new paths that are made by these deposited clusters become more and more likely to lie in parallel with existing paths as the coverage increases. The conductivity begins to diverge at a coverage of $p = p^*$, where $p^* = 0.79$ is the critical coverage for the CRR case, calculated in section 5.2.2.At high coverages like this, it is possible that the reduced-coverage system will continue to span between the terminals.

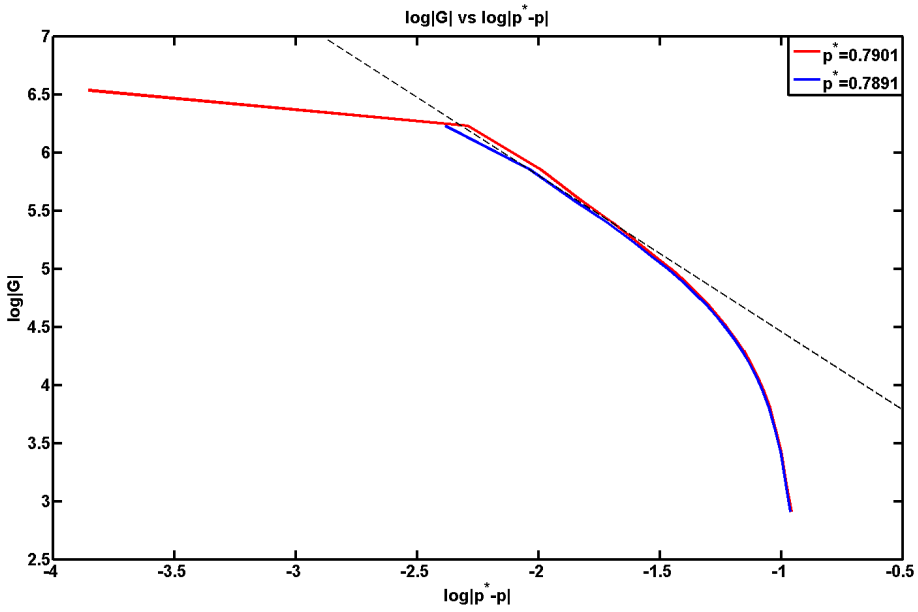


Figure 5.17: The conductivity of the 'constant radius reduction' network, graphed against $p^* - p$ for the 400x400 systems. The data was averaged over 100 trials. The red line uses $p^* = 0.7901$, while the blue line uses $p^* = 0.789$ the sensitivity of the slope to the value of p^*.

As in figure 5.8, it is possible that the CRR model will produce power-law behaviour when p^* is used instead of p_c. If so, the data would produce a power-law relationship similar to equation 2.19. In this case, the relationship would be given by $G \propto (p^* - p)^{-s}$, where $s = t = 1.3$ in two dimensions. Figure 5.17, incorporating this adjustment to the value of p_c, demonstrates a slope that approaches the expected value of $s = 1.3$ as p increases and tends towards p^*. This linear region only appears to extend between $log_{10}|p^* - p| = -1.5$ to -2.3, corresponding with coverages $p = 0.76$ to 0.785 respectively. With this in mind, it appears that the random superconductor network-like nature of the CRR model only begins to

dominate as p approaches p^*. The data point at $p = 0.79$, closest to p^* (found on the graph at $log_{10}|p^* - p| = -3.8$), drops below the expected slope. It is possible that this could be due to the fact that systems near p^* are discarded if they continue to span the system following radius reduction, and so only the poorest conducting systems (the non-spanning systems) are selected at high coverages, biasing the coverages near p^* towards lower-than-expected conductivities.

5.5 Ramping the current

5.5.1 The AtA model

The ramping of current through the system was performed by following the process discussed in section 4.1. Each connection was given a local critical current I_{local} that was determined by its neck width w, so that $I_{local} = kw$. The constant k was simply a scaling parameter and was given the value of 0.5. The weak-link connections were treated as Josephson junctions, so that the resistance of a weaklink during iteration $n + 1$ was determined by the current I_n flowing through it on iteration n. If $I_n < I_{local}$, the link was considered superconducting and so the resistance R_{n+1} was set to $R_{n+1} = R_{SC}$, where $R_{SC} = 10^{-4}$ was the superconducting resistance. If $I_n > I_{local}$, then the link was considered normal, and was given a resistance determined by the overdamped Josephson junction equation as $R_{n+1} = R_{normal}\sqrt{1 - \frac{I_{local}^2}{I_n^2}}$. The normal-state resistance R_{normal} is given by $R_{normal} = (g)^{-1} = \frac{1}{w}$. When the current is being ramped down from a high value to a low value, the critical current of the entire system I_c is defined as being the current at which the conductivity of the system increases by a factor of 1000 relative to the value it held at the previous current. In other words, for current step n, $I_c = I_n$ if $G(I_n) = 1000 \times G(I_{n-1})$.

Current-ramp solutions almost always oscillated between two distinct sets of connection states. A sample of two-state solutions for various currents are pictured in figures 5.18 and 5.19, which present the state of individual connections in the network as unused (grey), superconducting (black) or resistive to some degree (red). Figure 5.18 presents the multiple states for two currents above the system's critical current I_c, while figure 5.19 presents the states for two currents below I_c. Note that the system is considered superconducting, so that the input current $I < I_c$, when it is possible to find a fully superconducting path through the network. Due to the lack of damping and adequate redistribution of current, this does not necessarily guarantee that the current through every individual connection is less than its local critical current, and so it remains possible for some connections to be sent slightly normal locally while still having the system as a whole stay superconducting.The hope was that the dynamic resistance of the weak links, due to their Josephson junction behaviour, would provide a damping mechanism to help redistribute the current and force the system towards a steady-state equilibrium solution. Unfortunately, the overdamped Josephson junction approach was not sufficient for this purpose, and the iterated solutions ended up oscillating between two different system states for all of the currents applied during the current ramp,

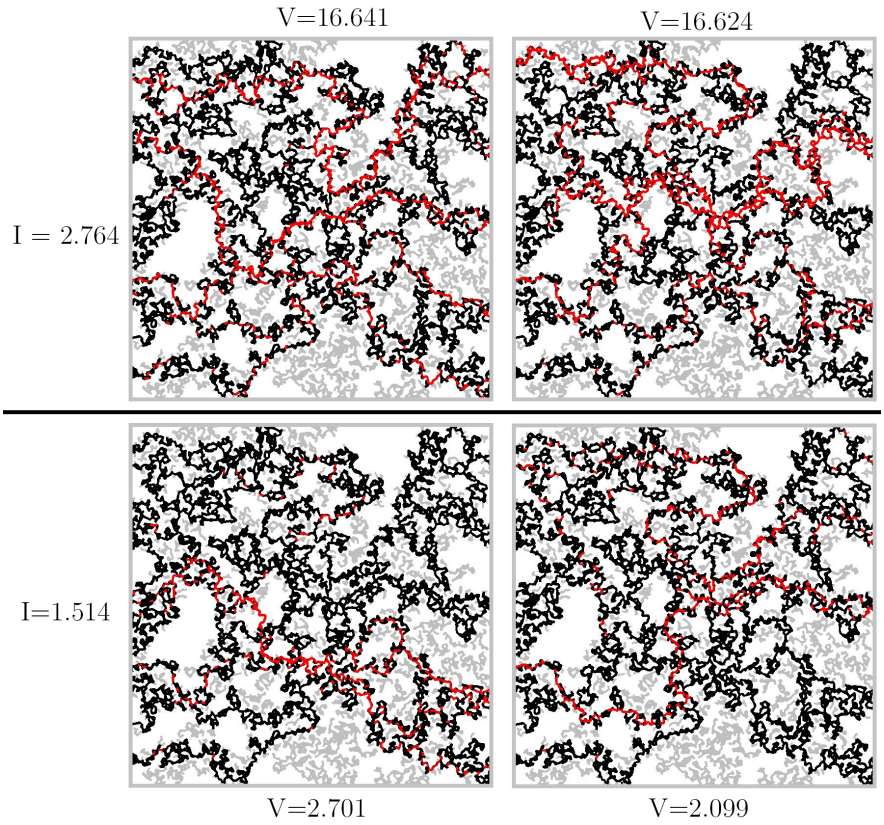


Figure 5.18: The states of the connections in the AtA system oscillate between two different sets of normal and superconducting connections as the current ramp is performed. This figure presents the sets of connection states associated with two different currents where $I > I_c$, in a 100x100 system at $p = 0.69$. The voltage difference across the system associated with each solution is indicated either above or below the system in question.

aside from the largest and smallest currents used.

5.5.2 V vs I

V-I curves and the critical current I_c of the system were calculated for each sample in turn. In the event of the system not reaching equilibrium during its iterations at a particular current, the measured voltage was taken to be the average of the voltages of the two oscillating states. The conductivity for each input current was calculated using the average voltage at the chosen current, and then compared with the conductivity at the previous current to check whether I_c had been found. Even with the oscillations, the system stayed either measurably normal or measurably superconducting across iterations, and so I_c was well defined and simple to recognise. Figure 5.20 demonstrates the typical progression of the voltage, for a range of currents near I_c and a selection of coverages, during a down-ramp of the current. The voltage drops in a step-like manner as various sections of connections turn from normal to superconducting, culminating in a final drop at $I = I_c$. Any further changes to the voltage due to oscillations are minor at best. Figure 5.21 features a much more finely grained current downramp in a 100x100

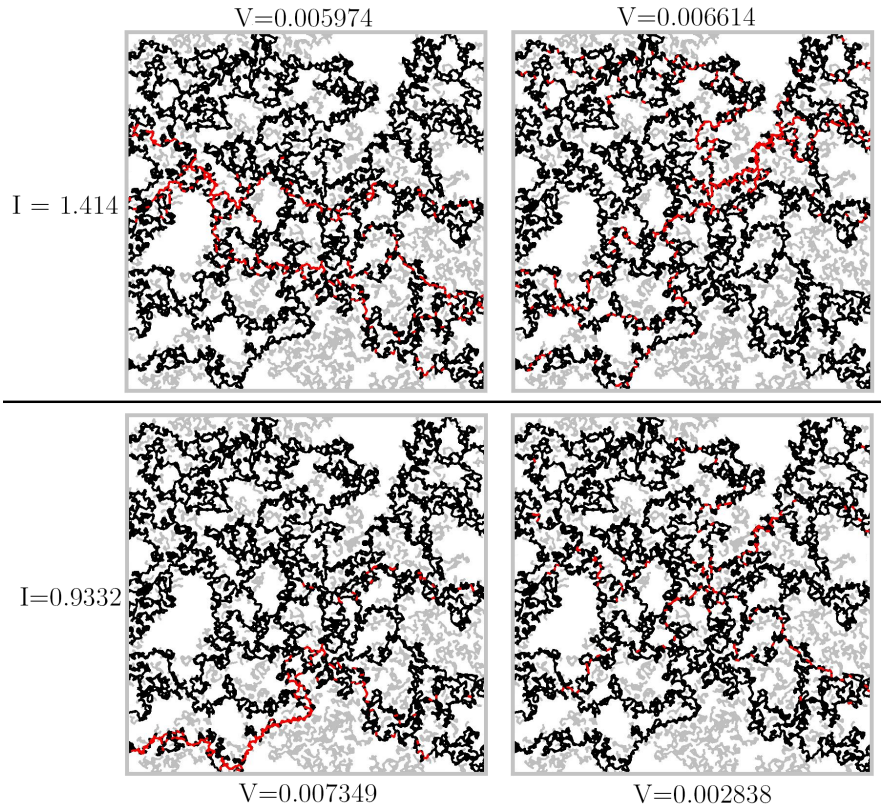


Figure 5.19: The states of the connections in the system oscillate between two different sets of normal and superconducting connections as the current ramp is performed. This figure presents the sets of connection states associated with two different currents where $I < I_c$, in a 100x100 system at $p = 0.69$. The voltage difference across the system associated with each solution is indicated either above or below the system in question.

system, and shows the sharp drop into superconductivity at I_c. This sharp drop occured in all system studied, and uniquely identified the critical current of the system.

Figure 5.22 presents the plot of $\log_{10} V$ vs $\log_{10}(I - I_c)$, with an additional line of slope $a = 2$ plotted for comparison with equation 2.18. In both cases, the expected slope of 2 appears to be reasonable almost exclusively within the range of I_c to $10I_c$. At higher currents well above I_c, after the vast majority of the backbone connections have turned normal, the few superconducting connections that are still changing to normal at this stage have little effect on the voltage across the system. At this point, even if not every single connection is normal, the relationship becomes Ohmic so that the exponent tends towards $\alpha = 1$. Close to I_c, the non-zero resistance of the superconducting connections prevents the voltage from tending towards zero as it should have done. The finite size effects that have influenced low coverages in the second moment and conductivity are also seen here. The lowest coverage shown, $p = 0.69$, also features the worst slope in the 200x200 system. The 400x400 system is much better, as the lowest coverage $p = 0.69$ is above the p-value of 0.688 that was found for the 400x400 system in section 5.2 and which indicates where the expected percolation exponents are lost. The linear slope of $\alpha = 2$ present a good fit of the data at current close to

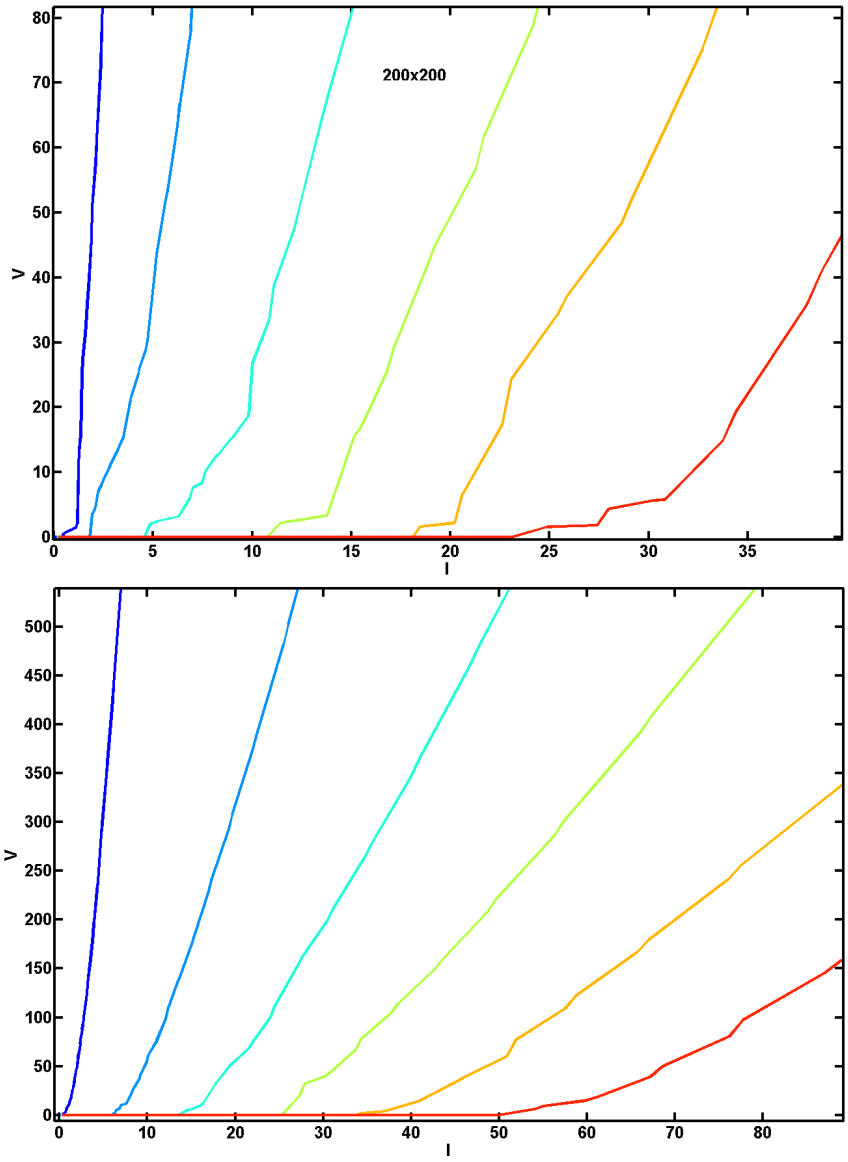


Figure 5.20: The voltages produced during a coarse-grained current down-ramp, for $p = 0.69$ to $p = 0.79$ in steps of 0.02 in 200x200 and 400x400 networks. The current-steps close to I_c are on the order of $0.3 - 0.5$ current units.

I_c, and is in reasonable agreement at these system sizes.

5.5.3 $\log_{10}(I_c)$ vs $\log_{10}|p - p_c|$

The last thing to check during the current ramp is the progression of I_c as the coverage is raised. As the coverage is increased, and more clusters are deposited into the system, we expect the percolating system to transition from the wire-like backbone near $p = p_c$ to a continuous two-dimensional film as p tends towards 1. Another way of viewing this is that we expect the narrowest part of the backbone to transition from a width on the scale of a single neck to a width that occupies the entire system. In the context of critical currents, this implies that the critical

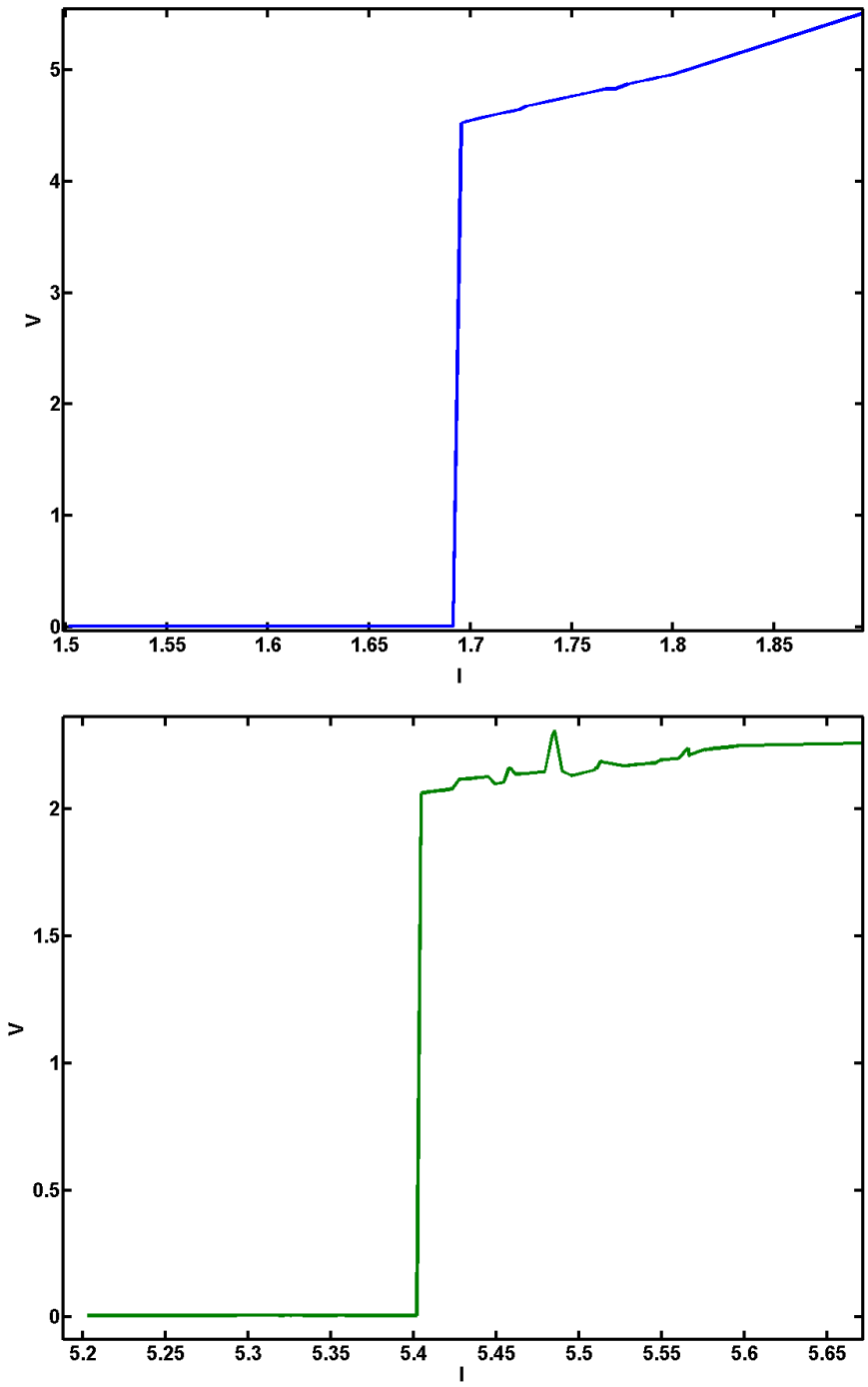


Figure 5.21: The voltages produced by a range of currents near I_c for $p = 0.69$ and $p = 0.75$ in a 100x100 system. These were computed during a fine-grained current downramp, where the current-steps near I_c are on the order of 0.01 current units. The variations of the voltage at comparable currents are essentially noise, due to the unusual oscillatory nature of the solutions.

current of the system should also rise as the percolating system provides more available paths, and so I_c of the system should be expected to scale as a power-law as well. The relevant power law is given in equation 2.17 by $I_c \propto (p - p_c)^v$, where $v = \frac{4}{3}$ in the Josephson case where the width of the constrictions is less

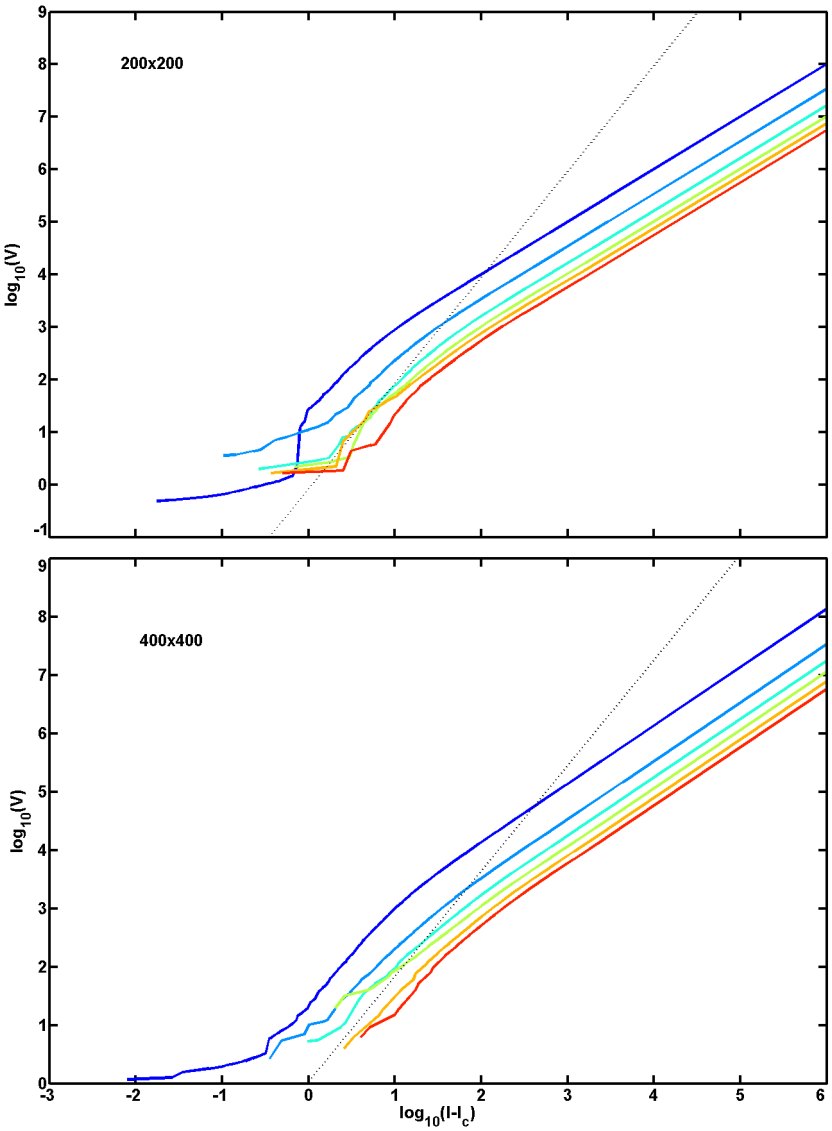


Figure 5.22: The voltage of the AtA network, graphed against the difference $I - I_c$ for two system sizes 200x200 and 400x400. Both system sizes are represented by a single typical system. A line with a gradient of 2 has been added to each plot for reference.

than the superconducting coherence length ξ_s so that $I_{local} \propto 1$, and $v = 1.821$ in the depairing case where the neck width is larger than ξ_s and so $I_{local} \propto w$, as discussed in section 4.1.

In figure 5.23, $\log(I_c)$ is graphed against $\log(p - p_c)$ for 200x200 and 400x400 systems, which have been averaged over 20 and 5 trials respectively. Again, the two system sizes are still under the influence of finite size effects and so, like the second moment (figure 5.5) and the conductivity (figure 5.14), the two or three coverages closest to p_c are not reliable. A slope of 1.82, associated with depairing as discussed by by Octavio et al. [62], has been plotted against both systems. The critical currents of both systems are in substantial agreement with the values expected.

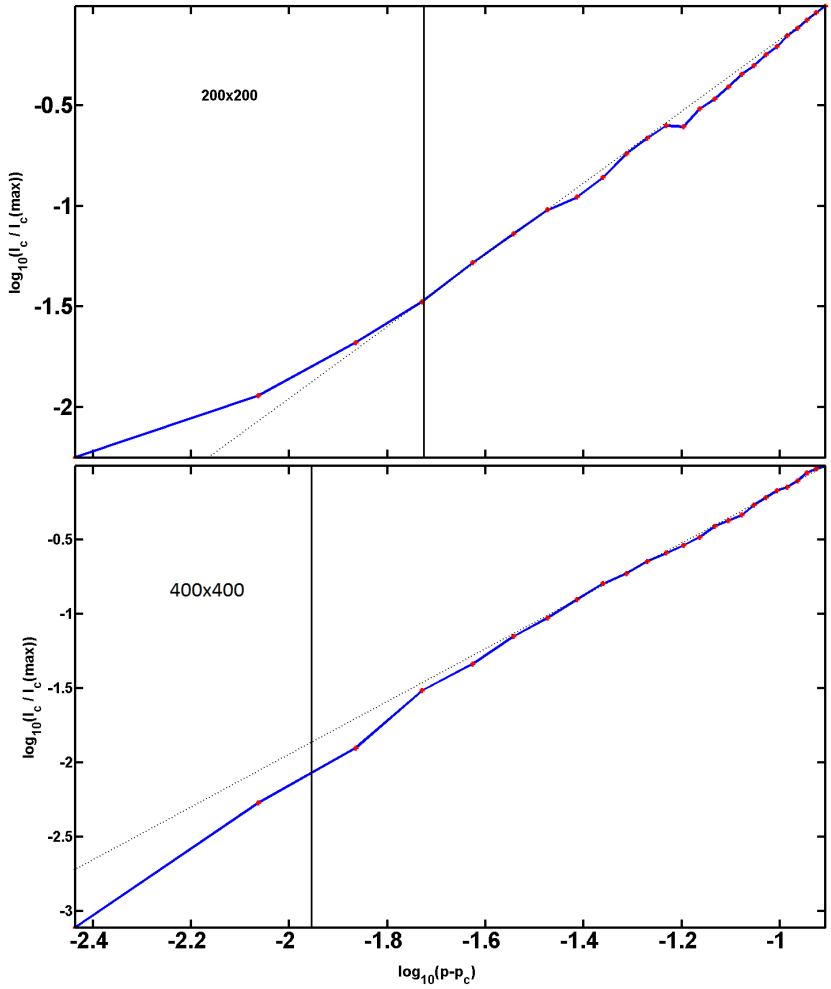


Figure 5.23: The critical current I_c, normalised by the highest I_c measured, graphed against the difference between the coverage p and p_c. Shown are two system sizes, 200x200 and 400x400, from the AtA model. The 200x200 system is averaged over 20 trials, while the 400x400 system is averaged over 5. A line with a gradient of 1.8 has been added to each plot for reference to the expected depairing exponent.

5.5.4 The CRR and CPR models

Both the voltage-current behaviour, and the relationship between the critical current I_c and the system coverage p, were investigated for the CRR and CPR models. The analysis was performed in the same way as it was for the AtA model. The value of $|p - p_c|$ at which finite size effects become important has been indicated on each I_c plot with a solid vertical line. Figure 5.24 presents both the voltages V measured at a series of input currents I during the current ramp, and I_c data as a function of $|p - p_c|$, for the CRR model. In comparison with the data for the AtA model presented in figure 5.22, the persistent slope of 1 across the range of currents is not promising. However, the slope of the $\log_{10}(V)$ plot at low coverages is not 1, and is beginning to look similar to the expected slope of 2 from reference [63]. This is due to the fact that at low coverages, a large proportion of all connections in the network are considered to be weak links by

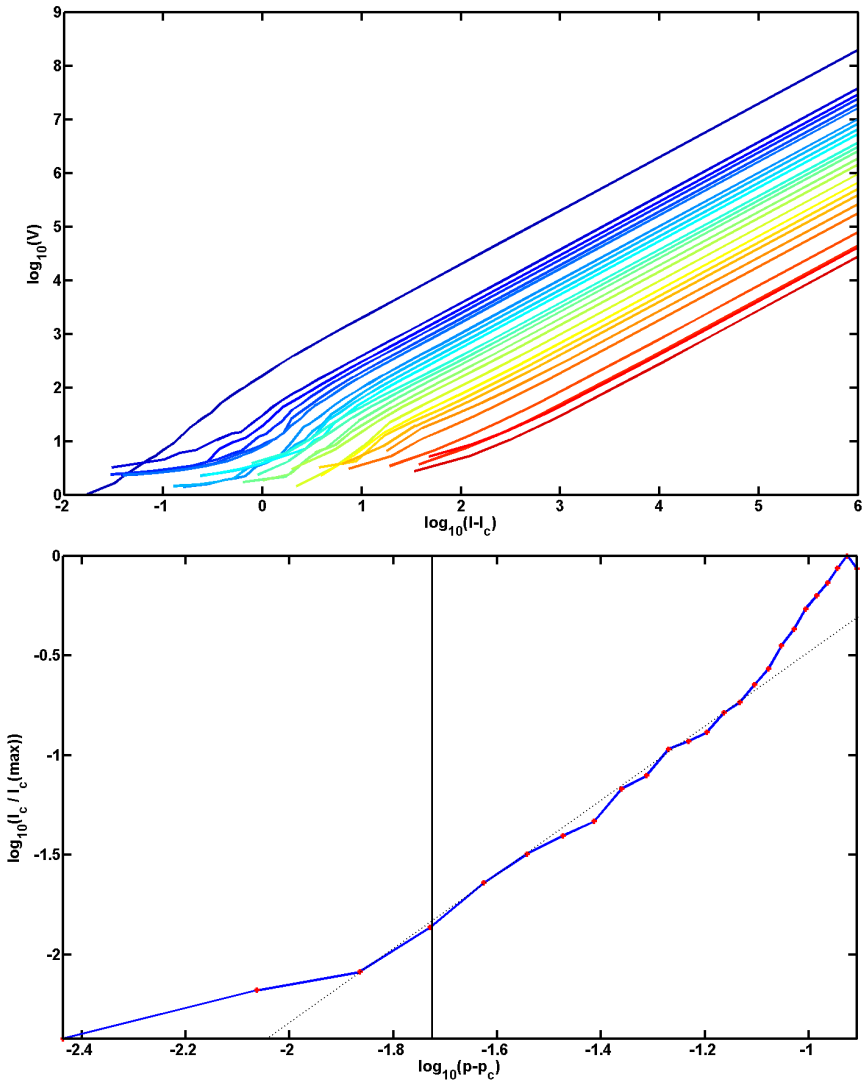


Figure 5.24: Graphs of voltage V against $I - I_c$, and the critical current I_c as function of $p - p_c$, are shown for a 200x200 CRR system. I_c is averaged over 10 trials. A slope of 1.8 has been added to the I_c plot for reference to the expected depairing exponent.

the CRR model, as seen in figure 5.11. At these coverages, the CRR systems are similar to the AtA systems in terms of the network of weak links produced. At higher coverages, however, the proportion of all connections that are weak links in the CRR system will decrease, and so we expect the CRR model to deviate further and further from the results produced by the AtA model as the coverage increase, due to the sheer lack of weak links involved in the network.

The I_c plot, on the other hand, matches the expected exponent of $v = 1.8$, which it shares with the AtA model as seen in figure 5.23. Aside from the finite size effects at the lowest coverages, the fit holds well until the coverage reaches $p \approx 0.76$, where the slope increases significantly. This is due to the increased capacity of the groups of 'strong' connections to carry and spread current. The groups of strong connections, with their average size demonstrated by the second moment in figure 5.7, are small until the higher coverages, where their size gets them involved in the distribution of current, spreading I over a large number of

connections and thus requiring that the critical current I_c is a high value to force all of the weak links involved into the normal state.

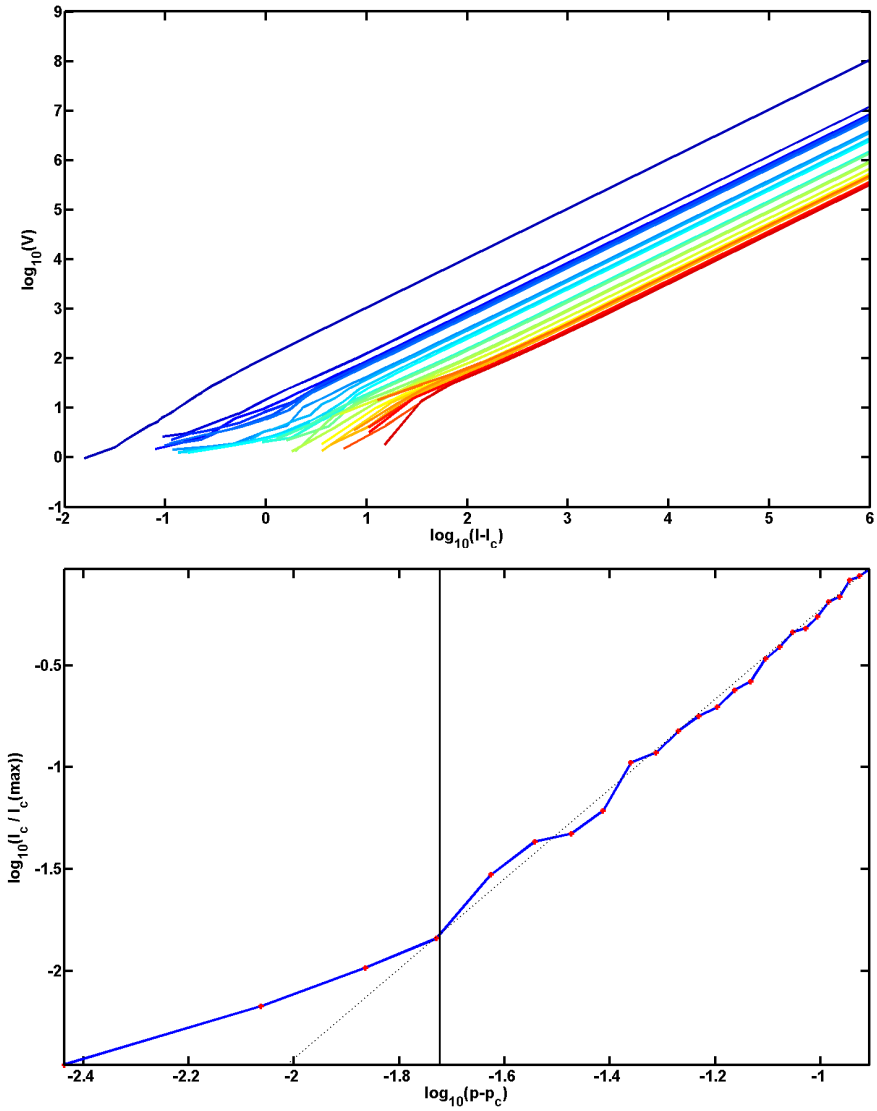


Figure 5.25: Graphs of voltage V against $I - I_c$, and the critical current I_c as function of $p - p_c$, are shown for a 200x200 CPR system. I_c is averaged over 10 trials. A slope of 1.8 has been added to the I_c plot for reference to the expected depairing exponent.

Figure 5.25 presents both the voltages V measured at a series of input currents I during the current ramp, and I_c data as a function of $|p - p_c|$, for the CPR model. The slopes of the $\log_{10}(V)$ vs $\log_{10}(I - I_c)$ plots are equal to 1 for the vast majority of currents and coverages, a situation already discussed for figure 5.24. However, for the CPR model, the number of weak-link connections is expected to increase significantly as the system coverage increases towards $p = 0.8$. As the number of weak-link connections at higher coverages gets larger, the system tends slowly towards replicating the AtA system. Because of this, we should see the expected exponent of $a = 2$ in the higher coverage plots. This is not observed in the figure.

The I_c plot featured in figure 5.25 features a strongly linear slope at values of

$|p - p_c|$ above the coverages affected by finite size effects. However, the slope being matched is $v = 2.1$, rather than the expected $v = 1.8$. This is due to the shifting nature of the weak-link threshold λ with coverage. As the coverage increases, not only does the number of deposited clusters (and thus, the number of available connections) increase, but the weak-link threshold λ increases so that more connections become weak links. This is important at I_c, where the critical current of the system is dictated by the critical currents of the weakest links along each parallel path between the terminals. As p goes up, not only does the number of these parallel paths increase, but the potential critical currents of the individual necks increase, as weak links with larger and larger neck sizes are considered. These effects compound to force the critical current I_c of the system higher and higher as the coverage increases, producing the observed deviation in the slope of the I_c plot in figure 5.25.

5.6 Model performance

The average times required by the AtA, CRR and CPR model to complete the two main algorithms used in this thesis, the conductance assignment function discussed in section 3.4 and appendix B, and the current ramp function discussed in section 4.1 and appendix C, are compared in figures 5.26 and 5.27 respectively. Both are extremely similar. The first, obvious take-away is that the AtA model takes far more time to compute than either of the two approximations, as would be expected. The interesting feature is that the CRR model times only begin to compare with or surpass the CPR times at $p > 0.76$, which is also the point at which the conductivity measured by the CRR model begins to deviate significantly from the expected relationships, as seen in figure 5.16. With that in mind, it appears that CPR is the faster program due to the fact that it processes the fewest number of weak-link connects up to $p = 0.75$ and, as seen in section 5.4.2, is much better than CRR at producing a power-law evolution of the conductivity, even with the slightly unusual slope.

5.7 Discussion

As seen in most of the system properties explored in the results, the finite size of the systems affected the results of system coverages near p_c. All three models were influenced simply by the geometry of the the initially deposited system, so this effect seems unavoidable for any individual system. The CPR and CRR models may have been affected to a larger extent, since they are predicated on selecting a substantially smaller number of weak links that the AtA model. This may have influenced their apparent suitability when compared with AtA for the same system sizes.

On the other hand, the resolution of the coarse-grained current ramp was adversely affected by increasing system size. The coarse ramp chose the next current during each iteration by selecting a current that would force 10% of the connections to go superconducting at each step. For larger system sizes, the 10%

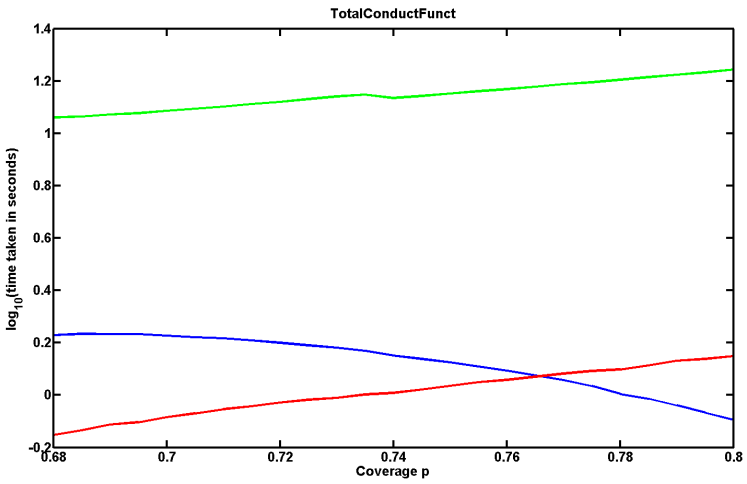


Figure 5.26: The time taken by the conductance function is displayed as a function of coverage for all three models: AtA in green, CPR in red and CRR in blue. The time data has been placed on a log scale for clarity. The times were averaged over a set of 10 attempts at each coverage.

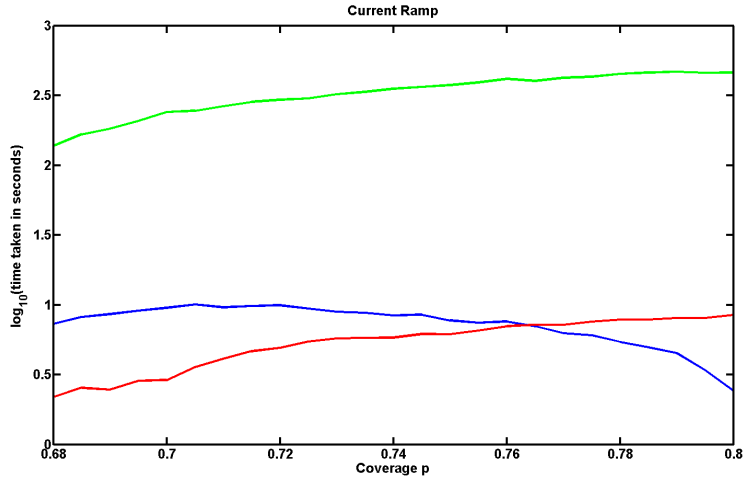


Figure 5.27: The time taken by the current ramp function is displayed as a function of coverage for all three models: AtA in green, CPR in red and CRR in blue. The time data has been placed on a log scale for clarity. The times were averaged over a set of 10 attempts at each coverage.

constitutes a larger number of connections, and so the visible effect of any individual connection going superconducting is obscured further and further. Around I_c, this became a problem as the current jumps were too large to accurately identify I_c. To this end, the high resolution version of the current ramp, which only turned one connection superconducting at each round, was required. However, this was also adversely affected by increasing system size, as it took far longer to work through the list of connections one-by-one as the number of connections increased with the system. For 200x200 and 400x400 system, the high-res ramp was prohibitively time-consuming. A more intelligent method or more computing power would be required to surmount this difficulty.

An additional problem with the current ramp was the difficulty with system switching, as mentioned in section 5.5. The original design of the current-ramp procedure required that the final system state from the previous current was used as the initial system state for the next current to introduce continuity and the possibility of hysteresis to the ramping routine. This was successful for down-ramps, but up-ramps had an issue where the initial oscillation between two major backbone paths, at low currents, would persist as the current was increased. At currents approaching I_c, if one of the two paths was measured as superconducting, the other path would be mostly normal due to having received the majority of the current in the previous iteration. This state of affairs would reverse on the next iteration, and the two paths would swap their states as the iterations continued. Of course, as the input current was increased, the previous state of connections was used to seed the new current iterations, and the problem was exacerbated, with the current travelling down the superconducting path rising higher and higher, and the current through the normal path remaining low. The system needed to reach a current value several orders of magnitude above the down-ramp I_c before the second, superconducting path was finally forced normal. This is not a physical scenario, and was due to the lack of damping available to the system to prevent the wild switching of paths.

The solution that was finally used to remedy this was to set the initial state of all connections as being either superconducting or normal depending on whether the current was being increased or decreased respectively. This isn't a perfect solution, as it implies that the system has no 'memory' of its previous states, whereas in reality, heating effects mean that it matters what the previous current was - if a connection goes normal at any point, it doesn't switch to the superconducting state again until it has reduced its temperature below T_c, current below I_c and excluded all penetrating magnetic fields. This concept is not currently available to the current-ramp algorithm. A better solution would be to introduce some form of damping mechanism, perhaps associated with heating, whereby individual connections will not switch from normal to superconducting instantly if they have been in the superconducting state previously during the iterations associated with the current being iterated. By introducing knowledge of previous states to the assignment of conductances during the ramping process, the system would hopefully converge towards a stable equilibrium value, rather than instantly switch between two wildly different states.

There are a few comments to make about the performance of the various models overall, and their physical interpretation. The brutish AtA model is obviously the best choice for producing the correct characteristics of the percolating networks, but at a significantly slower speed than the others by orders of magnitude. Another significant factor is the memory usage: every cluster that is deposited has to have its properties, and the connections it makes with neighbours, recorded. With a constant density of clusters across the system, if the system size is doubled (from 400x400 to 800x800, for example), then the area and number of clusters required is quadrupled. While the 400x400 system used above required perhaps a maximum of 1GB of memory for storage, an 800x800 system was attempted - while it took substantially longer, it could still fit in onboard memory. 1600x1600 was out of the question though, as it would've required a minimum of 16GB of

onboard memory, regardless of the time spent. Current-ramps for the 400x400 system already took around a day or two per trial, and so even the time use of the 800x800 system made it infeasible.

The CPR model appears to work well near p_c, but slows down significantly at higher coverages. If higher coverages near 1 had been used, it would have started approaching the AtA model in terms of time taken, as the reduction factor was forced lower and lower. CPR still provides reasonable results between $p = 0.68$ and 0.8. The slope of the system conductivity shown in figure 5.15 does not share the universal exponent t of the AtA system, which is slightly concerning but might be mitigated by the choice of p_r value. The CRR acts model more like an RSN network in both the second moment and progression of conductors described in sections 5.2 and 5.3 respectively. The lack of a power-law evolution of the conductivity is an issue, and CRR suffers a similar problem to the original tunnelling system at high coverages, where it begins to span the terminals. Given a range of coverages to check, however, the choice of the radius reduction value λ could easily mitigate these issues.

The AtA model is the most physically justified approach, as it treated each cluster as an integral part of the network regardless of whether the cluster is located in the middle of a blob or on the periphery. The fact that it takes the blobs as well as the weak links into account is a significant feature, particularly when high currents or other extreme events are involved. The CRR model is a sensible approximation at low input currents, as it focuses its attention exclusively on the weakest available connections. This should catch the vast majority of the individual link dynamics, as long as the current isn't high enough that the blobs should become resistive. For the CPR model, setting a constant p_r almost guarantees that the system will avoid spanning the system, due to the fact that the radius reduction factor λ is forced lower and approaches 0 as the coverage increases. This is beneficial for ramping the current, where the system is expected to have a critical current I_c where it appears resistive, regardless of coverage. By contrast, CRR does not guarantee that this is possible at high coverages, as groups can still span the system if λ is not small enough for the chosen coverages. The CPR approach does not correlate to anything physical for the percolating cluster networks but it does introduce random resistor network (RRN) statistics when all the weak links are activated.

Chapter 6

Conclusion

During the course of this thesis, several methods of introducing the concept of weak links into systems of percolating clusters were investigated. Two goals were established leading into the investigation. Firstly, we wanted to produce weak-link networks from spanning percolation systems where the exponents of the power-law relationships associated with percolating systems could be calculated accurately. Secondly, we wanted to select a weak-link approach that could calculate these exponents quickly.

AtA was the most brute-force method of choosing weak links, where every connection was considered to be weak, and so potentially resistive. This approach produced the best values for the percolation exponents for the conductivity of the fully-normal system t, the slope of the $\log_{10} V$ vs $\log_{10}(I - I_c)$ curves, and the critical current I_c of the system. However, it required a significant time investment to generate the dozens to trials combined to produce the final data, with weeks of simulation required to produce even the five trials worth of current ramp data shown in figures 5.22 and 5.23.

The CRR and CPR models were designed to focus on only the weakest links by establishing a distance threshold for selecting weak links. The two models were significantly faster than the AtA model at constructing and analysing the weak-link networks, as demonstrated in figures 5.26 and 5.27. However, both models struggle to present the exponents expected by the literature and delivered by the AtA model. Both systems produce errant conductivity exponents for fully-normal weak-link systems, with the CPR model producing a constant exponent of 1.6 rather than the slope of 1.3 associated with the AtA model, and the conductivity of the CRR model featuring significant curvature below $p = p^*$ and diverging towards large values for $p > p^*$. Neither the CPR or CRR models presented recognisable behaviour in the voltage at currents near I_c that could produce a recognisable exponent for equation 2.18. The exponent v associated with the evolution of the critical current I_c, as a function of $|p - p_c|$, featured the best result for the CRR and CPR models, with the CRR model producing an exponent of $v = 1.8$, a direct match for the exponent produced by the AtA case. The CPR model, on the other hand, over-estimated the value as $v = 2.1$.

These results answer the goal by saying that only the AtA model is capable of

genuinely representing the percolating systems being studied. It was the only one to produce accurate exponents across all of the measures it was considered with. However, the time taken for it to produce these exponents was significant. With this in mind, it is possible to make use of the much faster CRR model to locate the critical current I_c of the system as a function of coverage, which was the only power-law relationship that the CRR model could calculate well. However, the CRR model is still restricted to coverages significantly lower than its adjusted spanning coverage p^*. All models only produce suitable results at higher coverages than the coverages at which finite size effects significantly affect the statistics of the system, as demonstrated in section 5.2. With this in mind, the AtA model can be applied to calculate and predict the transport properties of a wide range of weak-link networks.

There is still a lot of progress to be made with the simulation of the experimental granular films. Firstly, the switching issues inherent in the current ramping process could be solved by the introduction of a suitable damping mechanism. The introduction of physical phenomena such as phase slips or heating effects could serve as a suitable damping process, and would allow for the possibility of hysteresis, as well as improving the speed of the convergence towards a single system state. Secondly, phase, magnetic fields and temperature are significant factors in the determination of the transport properties of superconductors. Introducing these three concepts into the model would allow for a far larger range of physical effects to be observed. Thirdly, the simulations discussed in this thesis have been the subject of noticable finite size effects. Transporting the current set of algorithms to a high-performance computing system would allow for the simulation of much larger system sizes, to minimise the effects attributed to the finite size of the networks and allow for the investigation of the percolating networks at coverages much closer to p_c.

Appendix A

Deposit_Particles.m

```
breaklines
function Particles=Deposit_Particles(coverage,systemsize,s,old)

D_average=1;
r=D_average/2;
MeanArea=pi.*r.^2;

L=systemsize*D_average;
AspectRatio=[1,1];
SurfaceSize=L*AspectRatio;
SurfaceArea=SurfaceSize(1)*SurfaceSize(2);

if 0<coverage && coverage<1
    TotalCoverage=-log(1-coverage);
    NumberOfParticles=round(TotalCoverage*SurfaceArea/MeanArea);

NumberToAdd=NumberOfParticles-old.NumberOfParticles;
if NumberToAdd < 0
    disp('The calculated number of particles was less than the ...
        previous coverage! No particles will be added!')
    NumberToAdd = 0;
end
elseif coverage <= 0 || coverage >= 1
    fprintf('Choose a value for the first layer coverage between ...
        0 and 1 exclusive.\n')
    return
end


h = 2*r;
xi = -L/2:h:L/2; nx = numel(xi);
yi = -L/2:h:L/2; ny = numel(yi);

dx = h;
dy = h;

particlespercell = NumberToAdd*h.^2/SurfaceArea;

n1 = poissrnd(particlespercell,nx-1,ny-1);
ntot = sum(sum(n1));

```

```
x=complex(zeros(1,ntot+old.NumberOfParticles));
R=zeros(1,ntot+old.NumberOfParticles);
x(1:old.NumberOfParticles)=old.x+1i.*old.y;
R(1:old.NumberOfParticles)=old.R;

if isempty(old.idx)
    idx = cell(nx-1,ny-1);
else
    idx=old.idx;
end
runningtotal = old.NumberOfParticles;

for i = 1:nx-1
    for j = 1:ny-1

        n = n1(i,j);
        xlocal = (xi(i) + dx*rand(1, n)) + ...
            1i * (yi(j) + dy*rand(1, n));
        rlocal = r;
        x(runningtotal+1:runningtotal+n) = xlocal;
        R(runningtotal+1:runningtotal+n) = rlocal;
        idx{i, j} = [idx{i, j},(1:n) + runningtotal];
                                runningtotal = runningtotal + n;
    end
end

NumberOfParticles = runningtotal;
if NumberOfParticles==0
    error('Not enough particles. Try a higher coverage.')
end

ActualTotalCoverage=(sum(pi.*R.^2))/(L^2);


Particles.x = real(x);
Particles.y = imag(x);
Particles.R=R;
Particles.L=L;
Particles.Coverage=coverage;
Particles.NumberOfParticles=NumberOfParticles;
Particles.TotalCoverageSpec=TotalCoverage;
Particles.TotalCoverageActual=ActualTotalCoverage;
Particles.gridoccupants = idx;

initiallength=10000;
[I, J]=deal(zeros(initiallength,1));
entrycounter=0;

for i = 1:nx-1
    for j = 1:ny-1

        pairs = [i, j; i+1, j-1; i+1, j; i+1, j+1; i, j+1];

        for k = 1:size(pairs, 1)
           try
                [ii, jj] = meshgrid(idx{i, j}, idx{pairs(k, 1), pairs(k, 2)});
                if (entrycounter+numel(ii))>length(I)
                    I=[I;zeros(length(I),1)];
```

```
                    J=[J;zeros(length(J),1)];
                end

                I(entrycounter+1:entrycounter+numel(ii)) = ii(:);
                J(entrycounter+1:entrycounter+numel(jj)) = jj(:);
                entrycounter=entrycounter+numel(ii);
            catch
            end
        end
    end
end
I(entrycounter+1:end)=[];J(entrycounter+1:end)=[];

extremum = R(I) + R(J);
iTouching = abs(x(I) - x(J)) < extremum;

Particles.oldI=I;
Particles.oldJ=J;
I = I(iTouching);
J = J(iTouching);
Particles.touchingpointsI = I;
Particles.touchingpointsJ = J;

[p,r,Group,ngroups] = FindGroups(I,J,x,[]);

Particles.Group=Group;
Particles.ngroups=ngroups;
Particles.Strength(1:NumberOfParticles,:)=1;
Particles.p=p;
Particles.r=r;

Strength(1:max(Group),:)=0;

for i=1:max(Group)
    Strength(i)=sum(Group==i);
    if Strength(i)==0
        error('One of the groups has no particles in it!...
                                Check the code please.')
    end
end


iso=sum(Group==0);
Strength(end+1:end+iso)=1;

Particles.Strength=Strength;
Particles.NumberOfGroups=max(Group)+sum(Group==0);

Ninf=max(Particles.Strength);
InfGroup=find(Particles.Strength==Ninf);
if any(size(InfGroup)~=[1 1])
    InfGroup=InfGroup(1);
    fprintf('There is more than one ''largest'' group in this trial\n')
end

Groupmax.x=Particles.x(Particles.Group==InfGroup);
Groupmax.y=Particles.y(Particles.Group==InfGroup);
Groupmax.R=Particles.R(Particles.Group==InfGroup);
```

```
157 Particles.Groupmax=Groupmax;
158
159 end
```

Appendix B

TotalConductInfo.m

```
breaklines
function Particles = TotalConductInfo(Particles, correlation, pmin,...
alltunnelconduct, voltage, allorone,Gcut)

spanningonly=0;

EdgeLengthThreshold = 30;
if pmin==0
    RadiusReduction=0.85;
else
    RadiusReduction=sqrt((log(1-pmin))/(log(1-Particles.Coverage)));
end

if alltunnelconduct==1
    RadiusReduction=1;
elseif alltunnelconduct==2||alltunnelconduct==3
    EdgeLengthThreshold = 0;
    if alltunnelconduct==3
        RadiusReduction=0.00001;
    end
end
Particles.RadiusReduction=RadiusReduction;

Group=Particles.Group;
ngroups=Particles.ngroups;
p=Particles.p;
r=Particles.r;
I=Particles.touchingpointsI;
J=Particles.touchingpointsJ;


x=Particles.x+1i*Particles.y; R=Particles.R;

if alltunnelconduct<2
    [Du,Sig_1,Gsumall,Eg,~,~]= FindMinDistance(x,p,r,Group,R,correlation,...
                EdgeLengthThreshold,ngroups,1);
else
    [Du,Sig_1,Gsumall,Eg]=deal([]);
end
Particles.D_1 = Du;
Particles.TunnelEdges = Eg;
```

```
eval...
  (sprintf('Particles.Sig_1.Correlation\%g=Sig_1;',round(100*correlation)));
eval...
(sprintf('Particles.Sig_all.Correlation\%g=Gsumall;',round(100*correlation)));

useweaklinks=0;
Particles.SpanningCluster=useweaklinks;

L = Particles.L;
samplesize = L/2;

leftlead = (Particles.x-Particles.R)<=-samplesize;
leftgroups = Group(leftlead);
leftgroupsunique = unique(leftgroups);


rightlead = (Particles.x+Particles.R)>=samplesize;
rightgroups = Group(rightlead);
rightgroupsunique = unique(rightgroups);
if (intersect(leftgroupsunique,rightgroupsunique)~=0)
    useweaklinks=1;
    Particles.SpanningCluster=useweaklinks;
    if spanningonly==1
        return
    end
elseif spanningonly||alltunnelconduct>1
    disp('The system does not span the terminals, and ...
                tunnelling is disabled. Aborting.')
    return

end

Smalllines=(abs(x(I) - x(J)) < RadiusReduction*(R(I)+R(J)));

I1=I(Smalllines); J1=J(Smalllines);
[q,s,newGroup,tempngroups] = FindGroups(I1,J1,x,[]);

[~,~,wGsumall,Realweaklinks,numconnections,TouchingSeparations]=...
FindMinDistance(x,q,s,newGroup,R,correlation,EdgeLengthThreshold,tempngroups,2);

Particles.TouchingSeparations=TouchingSeparations;

tempy=sparse(tempngroups,tempngroups,0);

if ~isempty(Gsumall)
tempy(newGroup(Eg(:,1)),newGroup(Eg(:,2)))=...
                                        Gsumall(Group(Eg(:,1)),Group(Eg(:,2)));
end
newGsumall=sparse(wGsumall+tempy);

Particles.newGroup=newGroup;
eval(sprintf('Particles.newSig_all.Correlation\%g=newGsumall;',...
                                                round(100*correlation)));
Particles.weaklinksI = Realweaklinks(:,1);
Particles.weaklinksJ = Realweaklinks(:,2);
Particles.NumIntergroupConnections=numconnections;

V = voltage;
```

```
99
100 NotIsoGroups=max(Group);
101 NumberOfIsoParticles=sum(Particles.Group==0);
102 IsoGroupNumbers=NotIsoGroups+1:NotIsoGroups+NumberOfIsoParticles;
103 Group(Group==0)=IsoGroupNumbers;
104
105
106 newleftgroups = newGroup(leftlead);
107 newleftgroupsunique = unique(newleftgroups);
108
109
110 newrightgroups = newGroup(rightlead);
111 newrightgroupsunique = unique(newrightgroups);
112
113 if useweaklinks==1
114     Particles.spanningroups=intersect(leftgroupsunique,rightgroupsunique);
115
116     Group=newGroup;
117     leftgroupsunique = newleftgroupsunique;
118     rightgroupsunique = newrightgroupsunique;
119     if (intersect(leftgroupsunique,rightgroupsunique)~=0)
120
121         disp('There are no weak links separating the left and...
122                                 right hand sides of the spanning cluster!');
123
124         Particles.SpanningCluster=2;
125         return
126     end
127 end
128
129 temp=correlation*100;
130 str=[];
131 if useweaklinks==1
132     str='new';
133 end
134
135 if allorone||useweaklinks
136     conductancepairs = eval...
137                 (sprintf('Particles.\%sSig_all.Correlation\%d',str,temp));
138 else
139     conductancepairs = eval...
140                 (sprintf('Particles.\%sSig_1.Correlation\%d',str,temp));
141 end
142
143 conductancepairs(isnan(conductancepairs))=0;
144 conductancepairs(Gcut>conductancepairs(conductancepairs>0))=0;
145
146 grouplist = unique(Group);
147 numgroups = length(grouplist);
148 numleftgroups= length(leftgroupsunique);
149 numrightgroups = length(rightgroupsunique);
150
151 conductnet = -1*(conductancepairs + conductancepairs.');
152 clear conductancepairs;
153
154 badgroups = find(~any(conductnet,2));
155
156 conductnetoriginal = conductnet;
```

```
conductnet(badgroups,:)=[];
conductnet(:,badgroups)=[];

grouptogroup = 1:numgroups;
for i = 1:length(badgroups)
    temp1 = find(grouptogroup==badgroups(i));
    grouptogroup(temp1:end)=grouptogroup(temp1:end)+1;
end

grouptogroup(end-length(badgroups)+1:end)=[];

[~,~,group2,~] = FindGroups([],[],...
zeros(numgroups-length(badgroups),1),conductnet);

newbadgroups = zeros(1,numgroups); badgroupscounter=0;
for i = 1:max(group2)
    lia1=ismember(leftgroupsunique,grouptogroup(group2==i));
    lia2=ismember(rightgroupsunique,grouptogroup(group2==i));
    if sum(lia1)==0 || sum(lia2)==0
        newbadgroups(badgroupscounter+1:badgroupscounter+...
            numel(grouptogroup(group2==i)))= grouptogroup(group2==i);
        badgroupscounter=badgroupscounter+numel(grouptogroup(group2==i));
    end
end
newbadgroups(badgroupscounter+1:end)=[];
newbadgroups=sort(newbadgroups);

badgroups = sort([badgroups;newbadgroups.']);
if length(badgroups)==numgroups
   disp('No groups remaining! No meaningful conduction...
        between electrodes, so a solution is not possible.');
   return
end

conductnet = conductnetoriginal;
conductnet(badgroups,:)=[];
conductnet(:,badgroups)=[];
clear conductnetoriginal;


numgroupsold = numgroups;
numgroups = numgroups-length(badgroups);

numleftgroupsold = numleftgroups;
leftgroupsuniqueold = leftgroupsunique;
leftgroupsunique(ismember(leftgroupsunique,badgroups))=[];
numleftgroups = length(leftgroupsunique);

numrightgroupsold = numrightgroups;
rightgroupsuniqueold = rightgroupsunique;
rightgroupsunique(ismember(rightgroupsunique,badgroups))=[];
numrightgroups = length(rightgroupsunique);

grouptogroup = 1:numgroupsold;
for i = 1:length(badgroups)
    temp1 = find(grouptogroup==badgroups(i));
    grouptogroup(temp1:end)=grouptogroup(temp1:end)+1;
end
```

```
215
216 grouptogroup(end-length(badgroups)+1:end)=[];
217
218 allslots=numgroups+numleftgroups+numrightgroups;
219
220 Gmatrix=sparse(allslots,allslots,0);
221
222 identmatrix=sparse(1:numgroups,1:numgroups,ones(numgroups,1));
223
224 conductnet(identmatrix==1)=-1*sum(conductnet,2);
225
226 [netrow,netcol]=find(conductnet~=0);
227 Gmatrix(netrow,netcol)=conductnet(netrow,netcol);
228
229 for i = 1:numleftgroups
230     Gmatrix(grouptogroup==leftgroupsunique(i),numgroups+i)=1;
231     Gmatrix(numgroups+i,grouptogroup==leftgroupsunique(i))=1;
232 end
233 for i = 1:numrightgroups
234     Gmatrix(grouptogroup==rightgroupsunique(i),numgroups+numleftgroups+i)=-1;
235     Gmatrix(numgroups+numleftgroups+i,grouptogroup==rightgroupsunique(i))=1;
236 end
237
238 Voltage = zeros(allslots,1);
239
240 for i = 1:numleftgroups
241     Voltage(numgroups+i)=V;
242 end
243
244 disp('Solving the matrix');
245 solution = Gmatrix\Voltage;
246 solutiontrim = solution;
247 solutiontrim(abs(solutiontrim) <=1e-14)=0;
248
249
250 conductance = sum(abs(solutiontrim(numgroups+1:...
251 numgroups+numleftgroups)))/abs(V);
252 Particles.conductance = conductance;
253
254 Particles.leftgroups = leftgroupsunique;
255 Particles.rightgroups = rightgroupsunique;
256
257 if ~isempty(badgroups)
258     Particles.leftgroupsold = leftgroupsuniqueold;
259     Particles.rightgroupsold = rightgroupsuniqueold;
260     Particles.numgroupsreduced = numgroups;
261     Particles.badgroups = badgroups;
262
263     solutionfull = NaN(numgroupsold+numrightgroupsold+numleftgroupsold,1);
264     solutionfull(grouptogroup)=solutiontrim(1:numgroups);
265     [~,lwhere1]=ismember(leftgroupsunique,leftgroupsuniqueold);
266     [~,lwhere2]=ismember(rightgroupsunique,rightgroupsuniqueold);
267
268     solutionfull(numgroupsold+lwhere1)=...
269                 solutiontrim(numgroups+1:numgroups+numleftgroups);
270
271     solutionfull(numgroupsold+numleftgroupsold+lwhere2)=...
272                 solutiontrim(numgroups+numleftgroups+1:...
```

```
                numgroups+numleftgroups+numrightgroups);

    Particles.solutionfull=solutionfull;

else if ~isnan(conductance)&&conductance>0 &&conductance<1E20
    Particles.solutionfull = solutiontrim;
    end
end
Particles.solution = solutiontrim;

end
```

Appendix C

CurrentRamp.m

```
breaklines
function Particles = CurrentRamp(Particles, correlation,...
Icritfactor, currents,fullsave,upordown)

Icmode=0;
gwidthorconst=0;
jjorstep=0;
highres=0;
if highres
    highcovmax=5.9;highcovmin=5.5;
    lowcovmax=1.7;lowcovmin=1.4;
    if strcmp(upordown,'down')
        if Particles.Coverage>0.74
            currents(1)=highcovmax+0.1;
        elseif Particles.Coverage>0.68
            currents(1)=lowcovmax+0.1;
        end
    else
        if Particles.Coverage>0.74
            currents(1)=highcovmin+0.00001;
        elseif Particles.Coverage>0.68
            currents(1)=lowcovmin-0.1;
        end
    end
end


maxiteration=30;

normfactor=10^(-5);
supercondfactor=10^4;

spancluster=Particles.SpanningCluster;
temp=correlation*100;

if spancluster==2
    disp('There are no weak links separating the left and...
                right hand sides of the spanning cluster!')
    return
elseif spancluster==1
    conductors=eval(sprintf('Particles.newSig_all.Correlation\%d',temp));
```

```
    Group=Particles.newGroup;
    ngroups=max(Group);
else
    conductors = eval(sprintf('Particles.Sig_all.Correlation\%d',temp));
    Group=Particles.Group;
    ngroups=max(Group);
end
[conductrow,conductcol]=find(abs(conductors)>10^(-14));
conductorindex=sub2ind(size(conductors),conductrow,conductcol);

conductors=sparse(conductrow,conductcol,conductors(conductorindex),...
ngroups,ngroups,length(conductrow));

conductors=conductors*normfactor;

rightgroupsunique = Particles.rightgroups;
leftgroupsunique  = Particles.leftgroups;

badgroups=Particles.badgroups;

conductnet=conductors+conductors.';
clear conductors;
conductnet(badgroups,:)=[];
conductnet(:,badgroups)=[];
if isfield(Particles,'NumIntergroupConnections')
    numconn=Particles.NumIntergroupConnections;
    numconn(badgroups,:)=[];
    numconn(:,badgroups)=[];
else
    numconn=logical(conductnet);
end


[condnetrow,condnetcol]=find(conductnet);
conductnetindex=sub2ind(size(conductnet),condnetrow,condnetcol);


if Icmode==0
    Icritical=sparse(condnetrow,condnetcol,Icritfactor*...
                conductnet(conductnetindex),length(conductnet),...
                length(conductnet),length(condnetrow));

elseif Icmode
    if ~exist('numconn');
        numconn=logical(conductnet);
    end
    Icritical=Icritfactor*numconn;
end

if gwidthorconst
    conductnet=numconn;
end

grouplist = unique(Group);
numgroups = max(grouplist)-length(badgroups);
numleftgroups= length(leftgroupsunique);
numrightgroups = length(rightgroupsunique);

```

```
grouptogroup = 1:max(grouplist);
for i = 1:length(badgroups)
    temp1 = find(grouptogroup==badgroups(i));
    grouptogroup(temp1:end)=grouptogroup(temp1:end)+1;
end

grouptogroup(end-length(badgroups)+1:end)=[];

matsize=numgroups+numrightgroups+2;
initialGmatrix = sparse(matsize,matsize,0);

identmatrix = sparse(1:matsize,1:matsize,ones(matsize,1));
oldconductnet=conductnet;
[temprow,tempcol,tempvalues]=find(conductnet);
clear conductnet
if strcmp(upordown,'up')
    tempvalues=supercondfactor*ones(length(tempvalues),1);

    initialconductnet=sparse(temprow,tempcol,tempvalues,...
                size(oldconductnet,1),size(oldconductnet,2));

else
    initialconductnet=oldconductnet;
end


tempGmatrix=sparse(temprow,tempcol,tempvalues,matsize,matsize);

for i = 1:numleftgroups
    tempGmatrix(grouptogroup==leftgroupsunique(i),numgroups+1)=...
                supercondfactor;
    tempGmatrix(numgroups+1,grouptogroup==leftgroupsunique(i))=...
                supercondfactor;
end
tempGmatrix=tempGmatrix-diag(sum(tempGmatrix,2));

for i = 1:numrightgroups
    tempGmatrix(grouptogroup==rightgroupsunique(i),numgroups+2+i)=-1;
    tempGmatrix(numgroups+2+i,grouptogroup==rightgroupsunique(i))=1;
end

tempGmatrix(numgroups+1,numgroups+2)=1;
tempGmatrix(numgroups+2,numgroups+2)=1;

initialGmatrix(1:size(tempGmatrix,1),1:size(tempGmatrix,2))=tempGmatrix;


Current = zeros(numgroups+numrightgroups+2,1);
voltbox=zeros(2*length(currents)-1,maxiteration);

[conductance,totalcount,currentsused] = deal(zeros(2*length(currents),1));
[solutions,changedconnections,changedconnectionstates]=...
deal(cell(2*length(currents),maxiteration));

if fullsave
    [superstatecopy,currentratio]=deal(cell(2*length(currents),2));
end

```

```
157 reductionvsthreshold=zeros(length(currents),1);
158 [normalconnectioncount,superconnectioncount,oscillating]=...
159 deal(zeros(2*length(currents),maxiteration));
160
161 Gmatrix=initialGmatrix;
162 tempconductnet=initialconductnet;
163
164 systemIc=zeros(length(currents),1);
165 newI=[currents(1);currents(1)];
166
167 for k=1:length(currents)
168     currentsused(2*k-1:2*k)=newI;
169
170     for m=1:2
171         iterationcount=0;
172         solution=[];
173         [superstate,~]=deal(sparse(numgroups,numgroups,0));
174         savestate=sparse(numgroups,numgroups,1);
175         Current(numgroups+2)=newI(m);
176
177         Gmatrix=initialGmatrix;
178         tempconductnet=initialconductnet;
179
180         while (any(any(superstate-savestate))) && (iterationcount<maxiteration)
181             iterationcount=iterationcount+1;
182
183             solution = Gmatrix\Current;
184
185             tempconductnet=Gmatrix(1:numgroups,1:numgroups);
186             tempconductnet(logical(sparse(1:numgroups,1:numgroups,...
187
188
189             tempindex=(abs(solution)<1e-16);
190             solution(tempindex)=0;
191             groupvoltages=solution(1:numgroups);
192
193             voltrow=temprow;voltcol=tempcol;
194             currentindex=find(tempconductnet);
195
196             clear voltagematrix; clear tempcurrentratio
197
198             voltagematrix=sparse(voltrow,voltcol,...
199                 abs(groupvoltages(voltrow)-...
200                 groupvoltages(voltcol)),numgroups,numgroups);
201
202             currentmatrix=voltagematrix.*tempconductnet;
203
204             tempcurrentratio=sparse(voltrow,voltcol,...
205                 currentmatrix(currentindex)./Icritical(currentindex),...
206                 numgroups,numgroups);
207
208             savestate=superstate;
209             clear superstate; clear normstate;
210
211             superstate=sparse(voltrow,voltcol,...
212                         tempcurrentratio(currentindex)<(1+10^(-10)),...
213                         numgroups,numgroups);
214
```

```
215             normstate=sparse(voltrow,voltcol,...
216                         tempcurrentratio(currentindex)>=(1+10^(-10)),...
217                         numgroups,numgroups);
218
219             [superrow,supercol]=find(superstate);
220             [normrow,normcol]=find(normstate);
221             if jjorstep
222
223                 tempconductnet2=sparse([superrow;normrow],...
224                 [supercol;normcol],...
225                 [supercondfactor*ones(length(superrow),1);...
226                 oldconductnet(normstate)],numgroups+1,numgroups+1);
227
228             else
229                 tempconductnet2=sparse([superrow;normrow],...
230                 [supercol;normcol],[supercondfactor*ones(length(superrow),1);
231                 oldconductnet(normstate)./(sqrt(1-...
232                 (1./tempcurrentratio(normstate).^2)))],...
233                 numgroups+1,numgroups+1);
234
235             end
236
237             for i = 1:numleftgroups
238                 tempconductnet2(grouptogroup==leftgroupsunique(i),...
239                 numgroups+1)=supercondfactor;
240
241                 tempconductnet2(numgroups+1,grouptogroup==...
242                 leftgroupsunique(i))=supercondfactor;
243             end
244             tempconductnet2=tempconductnet2-diag(sum(tempconductnet2,2));
245             Gmatrix(1:size(tempconductnet2,1),1:size(tempconductnet2,2))=...
246                 tempconductnet2;
247
248                 clear tempconductnet2
249
250             changedconnections{2*(k-1)+m,iterationcount}=...
251                 logical(sparse(voltrow,voltcol,...
252                 abs(superstate(currentindex)-savestate(currentindex)),...
253                 numgroups,numgroups));
254
255             [changedconnrow,changedconncol]=find(changedconnections...
256                 {2*(k-1)+m,iterationcount});
257
258
259             newstate=sparse(changedconnrow,changedconncol,...
260                 superstate(changedconnections{2*(k-1)+m,iterationcount}),...
261                 numgroups,numgroups);
262
263             changedconnectionstates{2*(k-1)+m,iterationcount}=newstate;
264             if iterationcount>1
265                 oscillatecheck=sum(sum(abs(changedconnections{2*(k-1)+m,...
266                         iterationcount}-changedconnections{2*(k-1)+m,...
267                         iterationcount-1})));
268                 if oscillatecheck==0
269                     oscillating(2*(k-1)+m,iterationcount)=1;
270                 else
271                     oscillating(2*(k-1)+m,iterationcount)=0;
272                 end
```

```
            else
                oscillating(2*(k-1)+m,iterationcount)=0;
            end

            voltbox(2*(k-1)+m,iterationcount)=abs(solution(numgroups+1));

            if fullsave
                solutions{2*(k-1)+m,iterationcount}=solution;
            elseif iterationcount==29
                solutions{2*(k-1)+m,1}=solution;
            end

            rawlist=tempcurrentratio(currentindex);
            ratiolist=sort(rawlist(rawlist>=(1+10^(-10))));
            superlist=sort(rawlist(abs(rawlist)<(1+10^(-10))));

            if strcmp(upordown,'up')
                if isempty(superlist)
                    superconnectioncount(2*(k-1)+m,iterationcount)=0.1;
                    if iterationcount>3
                        break
                    end
                else
                    superconnectioncount(2*(k-1)+m,iterationcount)=...
                                length(superlist);
                end
            end

            if ~isempty(ratiolist)
                normalconnectioncount(2*(k-1)+m,iterationcount)=...
                                length(ratiolist);
            elseif strcmp(upordown,'down')
                normalconnectioncount(2*(k-1)+m,iterationcount)=0.1;
                    if iterationcount>3
                        break
                    end
            end
            if fullsave&&iterationcount==29
                superstatecopy{2*(k-1)+m,1}=superstate;
                currentratio{2*(k-1)+m,1}=sparse(voltrow,voltcol,...
                                rawlist,numgroups,numgroups);
            end
        end

        if iterationcount==maxiteration
            conductance(2*(k-1)+m) = newI(m)/(0.5*(voltbox(2*(k-1)+m,...
                        iterationcount)+voltbox(2*(k-1)+m,iterationcount-1)));
        else
            conductance(2*(k-1)+m) = newI(m)/(voltbox(2*(k-1)+m,...
                        iterationcount));
        end

        if ((2*(k-1)+m)~=1)&&((conductance(2*(k-1)+m)/...
                conductance(2*(k-1)+m-1))>100)
            systemIc(2*(k-1)+m)=1;
        elseif ((2*(k-1)+m)>1)&&((conductance(2*(k-1)+m)/...
                conductance(2*(k-1)+m-1))<0.01)

```

```
            systemIc(2*(k-1)+m)=1;
        end

        if fullsave
            superstatecopy{2*(k-1)+m,2}=superstate;

            currentratio{2*(k-1)+m,2}=sparse(voltrow,voltcol,...
                        rawlist,numgroups,numgroups);

        elseif fullsave==0
            solutions{2*(k-1)+m,2}=solution;
        end

        clear tempcurrentratio changedconnections changedconnectionstates

        [changedconnections,changedconnectionstates]=...
                        deal(cell(2*length(currents),maxiteration));

        totalcount(2*(k-1)+m)=iterationcount;

        fprintf('Current \% d of \% d complete!\n',k,length(currents))
    end


    if strcmp(upordown,'up')
        oldItemp=max(newI);
        if isempty(superlist)
            break
        end
        if highres
            if (Particles.Coverage>0.74)&&min(newI)>highcovmin&&...
                                max(newI)<highcovmax

                reductionvalue=superlist(end);

            elseif (Particles.Coverage>0.68)&&min(newI)>lowcovmin&&...
                                max(newI)<lowcovmax

                reductionvalue=superlist(end);

            elseif (length(superlist)>20)&&(length(superlist)<...
                                floor(superconnectioncount(1,end)/100))

                reductionvalue=superlist(end);

            elseif length(superlist)<20
                reductionvalue=superlist(end);
            else
                reductionvalue=superlist(end+1-ceil(length(ratiolist)/10));
            end

            if (Particles.Coverage>0.74)&&(oldItemp<=highcovmin)&&...
                                ((oldItemp/reductionvalue)>highcovmin)

                oldI=highcovmin*reductionvalue;

            elseif (Particles.Coverage>0.68)&&(oldItemp>lowcovmax)&&...
                                ((oldItemp/reductionvalue)<lowcovmax)
```

```
389
390                 oldI=lowcovmin*reductionvalue;
391
392             else
393                 oldI=oldItemp;
394             end
395             if (Particles.Coverage>0.74)&&((oldI/(0.99999*reductionvalue))>...
396                                 highcovmax)
397
398                 break
399
400             elseif (0.74>Particles.Coverage)&&(Particles.Coverage>0.68)&&...
401                                 ((oldI/(0.99999*reductionvalue))>lowcovmax)
402
403                 break
404
405             end
406         else
407             reductionindex=ceil(0.1*(length(ratiolist)));
408             if reductionindex>length(superlist)
409                 reductionvalue=superlist(1);
410             else
411                 reductionvalue=superlist(end+1-reductionindex);
412             end
413             oldI=oldItemp;
414         end
415
416         if ~exist('oldI','var')
417             oldI=oldItemp;
418         end
419
420
421         if max(newI)>(10^6+1)
422             break
423         end
424     else
425         oldItemp=min(newI);
426         if isempty(ratiolist)
427             break
428         end
429         if highres
430             if (Particles.Coverage>0.74)&&min(newI)<highcovmax&&...
431                                 max(newI)>highcovmin
432
433                 reductionvalue=ratiolist(1);
434
435             elseif (Particles.Coverage>0.68)&&min(newI)<lowcovmax&&...
436                                 max(newI)>lowcovmin
437
438                 reductionvalue=ratiolist(1);
439
440             elseif (length(ratiolist)>20)&&(length(ratiolist)<...
441                                 floor(normalconnectioncount(1,end)/100))
442
443                 reductionvalue=ratiolist(2);
444
445             elseif length(ratiolist)<20
446
```

```
                reductionvalue=ratiolist(1);

            else

                reductionvalue=ratiolist(ceil(length(ratiolist)/10));

            end
        else
            reductionvalue=ratiolist(ceil(length(ratiolist)/10));
        end
        if strcmp(updown,'down')&&highres
            if (Particles.Coverage>0.74)&&(oldItemp>highcovmax)&&...
                                ((oldItemp/reductionvalue)<highcovmax)

                oldI=highcovmax*reductionvalue;

            elseif (Particles.Coverage>0.68)&&(oldItemp>lowcovmax)&&...
                                ((oldItemp/reductionvalue)<lowcovmax)

                oldI=lowcovmax*reductionvalue;

            else
                oldI=oldItemp;
            end
            if (Particles.Coverage>0.74)&&((oldI/(1.00001*reductionvalue))...
                                <highcovmin)

                break

            elseif (Particles.Coverage<0.74)&&(Particles.Coverage>0.68)&&...
                                ((oldI/(1.00001*reductionvalue))<lowcovmin)

                break

            end
        else
            oldI=oldItemp;
        end
        if ~exist('oldI','var')
            oldI=oldItemp;
        end

    end


    reductionvsthreshold(k)=reductionvalue-1;
    if strcmp(updown,'down')
        newI=[oldI/(0.99999*reductionvalue);oldI/(1.00001*reductionvalue)];
    else
        newI=[oldI/(1.00001*reductionvalue);oldI/(0.99999*reductionvalue)];
    end

end


colstoremove=logical(sum(voltbox)==0);
voltbox(2*k+1:end,:)=[];
voltbox(:,colstoremove)=[];
```

```
505 changedconnections(2*k+1:end,:)=[];
506 changedconnections(:,colstoremove)=[];
507 changedconnectionstates(2*k+1:end,:)=[];
508 changedconnectionstates(:,colstoremove)=[];
509 oscillating(2*k+1:end,:)=[];
510 oscillating(:,colstoremove)=[];
511 normalconnectioncount(2*k+1:end,:)=[];
512 normalconnectioncount(:,colstoremove)=[];
513 superconnectioncount(2*k+1:end,:)=[];
514 superconnectioncount(:,colstoremove)=[];
515 solutions(2*k+1:end,:)=[];
516 solutions(:,colstoremove)=[];
517
518 if fullsave
519     currentratio(2*k+1:end,:)=[];
520     currentratio(:,colstoremove)=[];
521     superstatecopy(2*k+1:end,:)=[];
522     superstatecopy(:,colstoremove)=[];
523 end
524
525 if currentsused(1)>currentsused(end)
526     currentsused(currentsused==0)=[];
527     currentsused=currentsused(end:-1:1);
528     systemIc(length(currentsused)+1:end)=[];
529     systemIc=systemIc(end:-1:1);
530     conductance(length(currentsused)+1:end)=[];
531     conductance=conductance(end:-1:1);
532     totalcount(length(currentsused)+1:end)=[];
533     totalcount=totalcount(end:-1:1);
534     reductionvsthreshold(length(currentsused)+1:end)=[];
535     reductionvsthreshold=reductionvsthreshold(end:-1:1);
536     normalconnectioncount=flipud(normalconnectioncount);
537     superconnectioncount=flipud(superconnectioncount);
538     oscillating=flipud(oscillating);
539     voltbox=flipud(voltbox);
540     changedconnections=flipud(changedconnections);
541     changedconnectionstates=flipud(changedconnectionstates);
542     solutions=flipud(solutions);
543
544     if fullsave
545         currentratio=flipud(currentratio);
546         superstatecopy=flipud(superstatecopy);
547     end
548 end
549
550 Particles.CurrentRamp.currents=currentsused;
551 Particles.CurrentRamp.systemIc=systemIc;
552 Particles.CurrentRamp.conductance=conductance;
553 Particles.CurrentRamp.NumberOfIterations=totalcount;
554 Particles.CurrentRamp.voltages=voltbox;
555 Particles.CurrentRamp.IvsIcDifference=reductionvsthreshold;
556 Particles.CurrentRamp.NormalConnectionCount=normalconnectioncount;
557 Particles.CurrentRamp.SuperConnectionCount=superconnectioncount;
558 Particles.CurrentRamp.OscillatingCheck=oscillating;
559 Particles.CurrentRamp.solutions=solutions;
560
561 if fullsave
562     Particles.CurrentRamp.currentratio=currentratio;
```

```
563         Particles.CurrentRamp.superstatecopy=superstatecopy;
564     end
565
566     end
```

Appendix D

FindMinDistance.m

```
breaklines
function [Du,Sig_1,Gsumall,Eg,connectionmatrix,TouchingSeparations]=...
FindMinDistance(x,p,r,Group,R,correlation,...
EdgeLengthThreshold,ngroups,tunnellingorweaklink)

initiallength=10000;
TouchingSeparations=cell(initiallength,2);
[Ig, Jg, Dg, Gsum,nconnects]=deal(zeros(1,initiallength));
Eg=zeros(initiallength,2);
Ecounter=0;
insertcounter=0;

dt = delaunayTriangulation(real(x(:)), imag(x(:)));
edges = dt.edges;

for i2 = 1:ngroups
    Indices_of_group = p(r(i2):r(i2+1)-1);
    Indices_out = logical(sum(ismember(edges, Indices_of_group), 2) == 1);

    E = edges(Indices_out, :);

    E2 = E;
    E2(ismember(E2, Indices_of_group)) = 0;
    neighbourgroup = Group(sum(E2, 2)');

    Radiisum=R(E(:,1))+R(E(:,2));
    Separation=abs(x(E(:,1))-x(E(:,2)));
    RadiiDiff=abs(R(E(:,1))-R(E(:,2)));

    length_of_edge = Separation-Radiisum;

    for j2 = unique(neighbourgroup)
        insertcounter=insertcounter+1;
        if insertcounter>length(Ig)
            Ig=[Ig,zeros(1,length(Ig))];
            Jg=[Jg,zeros(1,length(Ig))];
            Dg=[Dg,zeros(1,length(Ig))];
            Gsum=[Gsum,zeros(1,length(Ig))];
            nconnects=[nconnects,zeros(1,length(Ig))];
            TouchingSeparations{2*length(Ig),2}=[];
        end
```

```
41          Ig(insertcounter)=i2; Jg(insertcounter)=j2;
42
43          ij = find(neighbourgroup == j2);
44          [temp2, imin] = min(length_of_edge(ij));
45
46          nconnects(insertcounter)=numel(ij);
47
48          Dg(insertcounter)=temp2;
49
50          if tunnellingorweaklink==1
51              if length_of_edge(ij)<EdgeLengthThreshold
52                  Gsum(insertcounter)=sum(exp(-correlation*...
53                                  length_of_edge(ij)));
54                  Ecounter=Ecounter+1;
55                  if Ecounter>length(Eg)
56                      Eg=[Eg;zeros(length(Eg),2)];
57                  end
58                  Eg(Ecounter,:)=E(ij(imin),:);
59              else
60                  Gsum(insertcounter)=0;
61              end
62
63          elseif tunnellingorweaklink==2
64              SeparationW=Separation(ij);
65              RadiisumW = Radiisum(ij);
66              RadiiDiffW=RadiiDiff(ij);
67
68              Touching=logical(length_of_edge(ij)<=0);
69
70              TouchingSeparations{insertcounter,1}=SeparationW(Touching);
71              TouchingSeparations{insertcounter,2}=[i2 j2];
72
73              if sum(Touching)>0
74                  w=sqrt(RadiisumW(Touching).^2-SeparationW(Touching).^2);
75                  widthtotal=sum(w);
76                  Gsum(insertcounter)=1E5*widthtotal;
77                  if Ecounter+numel(ij)>length(Eg)
78                      Eg=[Eg;zeros(length(Eg),2)];
79                  end
80                  for k=1:numel(ij)
81                      Eg(Ecounter+k,:)=E(ij(k),:);
82                  end
83                  Ecounter=Ecounter+numel(ij);
84              else
85                  Gsum(insertcounter)=0;
86              end
87          elseif (tunnellingorweaklink~=1)&&(tunnellingorweaklink~=2)
88              disp('The conductance assignment in ''FindMinDistance''...
89                          has failed. Please check the code.');
90          end
91      end
92  end
93  Ig(insertcounter+1:end)=[];
94  Jg(insertcounter+1:end)=[];
95  Dg(insertcounter+1:end)=[];
96  Gsum(insertcounter+1:end)=[];
97  nconnects(insertcounter+1:end)=[];
98  Eg(Ecounter+1:end,:)=[];
```

```
99  TouchingSeparations(insertcounter+1:end,:)=[];
100
101 connectionmatrix=sparse([Ig,Jg],[Jg,Ig],[nconnects nconnects],ngroups,ngroups);
102
103 Gsumall = sparse(Ig, Jg, Gsum, numel(r)-1, numel(r)-1);
104 Gsumall(Gsumall(Gsumall~=0)<1E-15) = 0;
105 Gsumall = triu(Gsumall);
106
107 if tunnellingorweaklink==1
108     try
109         D = sparse(Ig, Jg, Dg, numel(r)-1, numel(r) - 1);
110     catch
111         disp('Creating a sparse matrix of distances has failed...
112                                 Check that the indices are all integers.');
113         return
114     end
115     D(D>EdgeLengthThreshold)=0;
116     Du = triu(D);
117
118     Sig_1 = Du;
119     temp2 = logical(Du);
120     G1 = exp(-Du(Du~=0)*correlation);
121     Sig_1(temp2)=G1;
122 else
123     Du=[]; Sig_1=[];
124 end
125
126 end
```

Appendix E

FindGroups.m

```
breaklines
function [p,r,Group,ngroups] = FindGroups(I,J,x,Network)

if isempty(Network)
    G = sparse(I, J, true, numel(x), numel(x));
    G = G + G.';
else
    G = logical(Network);
end

[p, ~, r, ~] = dmperm(G);

Group = zeros(size(x));
ngroups = numel(r) - 1;
for i1 = 1:ngroups
    Group(p(r(i1):r(i1+1)-1)) = i1;
end

end
```

Bibliography

[1] Dominic J Benford. Transition edge sensor bolometers for cmb polarimetry. *CMBpol Technology Whitepaper.* , 2008.

[2] René Reichel, Jim G Partridge, Alan DF Dunbar, Simon A Brown, Owen Caughley, and Ahmad Ayesh. Construction and application of a uhv compatible cluster deposition system. *Journal of Nanoparticle Research*, 8(3-4):405–416, 2006.

[3] ADF Dunbar, JG Partridge, M Schulze, and SA Brown. Morphological differences between bi, ag and sb nano-particles and how they affect the percolation of current through nano-particle networks. *The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics*, 39(3):415–422, 2006.

[4] Shawn Fostner, Richard Brown, James Carr, and Simon A. Brown. Continuum percolation with tunneling. *Phys. Rev. B*, 89:075402, Feb 2014.

[5] B Zeimetz and NA Rutter. Computer simulation of current percolation in polycrystalline high-temperature superconductors. *Superconductor Science and Technology*, 672:0–4, 2001.

[6] B Zeimetz, BA Glowacki, and JE Evetts. Application of percolation theory to current transfer in granular superconductors. *The European Physical Journal B*, 29(3):359–367, October 2002.

[7] Bernhard Zeimetz, Bartek a Glowacki, and Jan E Evetts. Resistor network model for simulation of current and flux percolation in granular coated conductors. *Physica C: Superconductivity*, 372-376:767–770, August 2002.

[8] D Stauffer and A Aharony. *Introduction To Percolation Theory.* Taylor & Francis, 1994.

[9] MA Lindeman, MB Anderson, SR Bandler, N Bilgri, J Chervenak, S Gwynne Crowder, S Fallows, E Figueroa-Feliciano, F Finkbeiner, N Iyomoto, et al. Percolation model of excess electrical noise in transition-edge sensors. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 559(2):715–717, 2006.

[10] SR Broadbent and JM Hammersley. Percolation processes I. Crystals and mazes. *Proc. Cambridge Philos. Soc*, pages 629–641, August 1957.

[11] M Sahini and M Sahimi. *Applications of percolation theory*. CRC Press, 1994.

[12] Madhavi Chand, Garima Saraswat, Anand Kamlapure, Mintu Mondal, Sanjeev Kumar, John Jesudasan, Vivas Bagwe, Lara Benfatto, Vikram Tripathi, and Pratap Raychaudhuri. Phase diagram of the strongly disordered s-wave superconductor nbn close to the metal-insulator transition. *Phys. Rev. B*, 85:014508, Jan 2012.

[13] NA Rutter, BA Glowacki, and JE Evetts. Percolation modelling for highly aligned polycrystalline superconducting tapes. *Superconductor Science and Technology*, 25, 2000.

[14] Sang Bub Lee and S Torquato. Pair connectedness and mean cluster size for continuum percolation models: Computer simulation results. *The Journal of Chemical Physics*, 89(10):6427–6433, 1988.

[15] A Coniglio. Cluster structure near the percolation threshold. *Journal of Physics A: Mathematical and General*, 3829, 1982.

[16] BJ Last and DJ Thouless. Percolation theory and electrical conductivity. *Physical Review Letters*, 27(December), 1971.

[17] Olaf Stenull and HK Janssen. Conductivity of continuum percolating systems. *Physical Review E*, 1:1–14, 2001.

[18] S Kirkpatrick. Classical transport in disordered media: scaling and effective-medium theories. *Physical Review Letters*, 27(2):1–4, 1971.

[19] JP Straley. Critical exponents for the conductivity of random resistor lattices. *Physical Review B*, 15(1), 1977.

[20] RB Pandey, D Stauffer, A Margolina, and J.G. Zabolitzky. Diffusion on random systems above, below, and at their percolation threshold in two and three dimensions. *Journal of Statistical Physics*, 34(3-4):427–450, 1984.

[21] B Watson and P Leath. Conductivity in the two-dimensional-site percolation problem. *Physical Review B*, 9(11):4893–4896, June 1974.

[22] M Sahimi, B D Hughes, L E Scriven, and H T Davis. Critical exponent of percolation conductivity by finite-size scaling. *Journal of Physics C: Solid State Physics*, 16(16):L521, 1983.

[23] S Feng, BI Halperin, and PN Sen. Transport properties of continuum systems near the percolation threshold. *Physical review. B, Condensed matter*, 35(1):197–214, January 1987.

[24] Peter Grassberger. Conductivity exponent and backbone dimension in 2-d percolation. *Physica A: Statistical Mechanics and its Applications*, 262(34):251 – 263, 1999.

[25] Markus Porto, Armin Bunde, Shlomo Havlin, and H Eduardo Roman. Structural and dynamical properties of the percolation backbone in two and three dimensions. *Phys. Rev. E*, 56:1667–1675, Aug 1997.

[26] JP Straley. Threshold behaviour of random resistor networks: a synthesis of theoretical approaches. *Journal of Physics C: Solid State Physics*, 2333, 1982.

[27] R Pike and H E Stanley. Order propagation near the percolation threshold. *Journal of Physics A: Mathematical and General*, 14(5):L169, 1981.

[28] M Sahimi. Finite-size scaling calculation of conductivity of three-dimensional conductor-superconductor networks at percolation threshold. *Journal of Physics C: Solid State Physics*, 355, 1984.

[29] JT Chayes, L Chayes, Daniel S Fisher, and T Spencer. Finite-size scaling and correlation lengths for disordered systems. *Physical review letters*, 57(24):2999, 1986.

[30] John Cardy. *Finite-size scaling*. Elsevier, 2012.

[31] A-MS Tremblay and J Machta. Finite-size effects in continuum percolation. *Physical Review B*, 40(7):5131, 1989.

[32] Paul Balister, Amites Sarkar, and Bla Bollobs. Percolation, connectivity, coverage and colouring of random geometric graphs. In Bla Bollobs, Robert Kozma, and Dezs Mikls, editors, *Handbook of Large-Scale Random Networks*, volume 18 of *Bolyai Society Mathematical Studies*, pages 117–142. Springer Berlin Heidelberg, 2008.

[33] ET Gawlinski and HE Stanley. Continuum percolation in two dimensions: Monte Carlo tests of scaling and universality for non-interacting discs. *Journal of Physics A: Mathematical and General*, 291, 1981.

[34] J Quintanilla, S Torquato, and R M Ziff. Efficient measurement of the percolation threshold for fully penetrable discs. *Journal of Physics A: Mathematical and General*, 33(42):L399, 2000.

[35] JF McCarthy. Continuum Percolation of Disks and the Random Lattice. *Physical Review Letters*, 58(21):2242–2244, 1987.

[36] WT Elam, AR Kerstein, and JJ Rehr. Critical Properties of the Void Percolation Problem for Spheres. *Physical Review Letters*, 52(17):1516–1519, 1984.

[37] BI Halperin, S Feng, and PN Sen. Differences between lattice and continuum percolation transport exponents. *Physical review letters*, 54(22), February 1985.

[38] S. Sadeghnejad, M. Masihi, and P.R. King. Dependency of percolation critical exponents on the exponent of power law size distribution. *Physica A: Statistical Mechanics and its Applications*, 392(24):6189 – 6197, 2013.

[39] Stephan Mertens and Cristopher Moore. Continuum percolation thresholds in two dimensions. *Physical Review E*, pages 1–6, 2012.

[40] Ali Dalafi, Tahoora Dalafi, Iraj Hadi, Sadollah Ebrahimi, and Mahmoud Jafari. Computer study of critical exponents in two-dimensional systems

of circular and sticklike nanoparticles. *Computational Materials Science*, 68(0):287 – 289, 2013.

[41] Jiantong Li and Mikael Östling. Percolation thresholds of two-dimensional continuum systems of rectangles. *Phys. Rev. E*, 88:012101, Jul 2013.

[42] V Sasidevan. Continuum percolation of overlapping disks with a distribution of radii having a power-law tail. *Physical Review E*, 88(2):022140, 2013.

[43] Sang Bub Lee. Multiple fragmentation of critical continuum percolation clusters. *Physica A: Statistical Mechanics and its Applications*, 393:480–488, January 2014.

[44] G Ambrosetti, I Balberg, and C Grimaldi. Percolation-to-hopping crossover in conductor-insulator composites. *Physical Review B*, 82(13):134201, October 2010.

[45] M Tinkham. *Introduction to Superconductivity*. Dover books on physics and chemistry. Dover Publications, 2004.

[46] H Kamerlingh Onnes. Further experiments with liquid helium. c. on the change of electric resistance of pure metals at very low temperatures etc. iv. the resistance of pure mercury at helium temperatures. In *KNAW, Proceedings*, volume 13, pages 1910–1911, 1911.

[47] H Kamerlingh Onnes. *Further experiments with Liquid Helium. G. On the Electrical Resistance of Pure Metals, etc. VI. On the Sudden Change in the Rate at which the Resistance of Mercury Disappears.* Springer, 1991.

[48] John Bardeen, Leon N Cooper, and J Robert Schrieffer. Theory of superconductivity. *Physical Review*, 108(5):1175, 1957.

[49] H Fröhlich. Theory of the superconducting state. i. the ground state at the absolute zero of temperature. *Physical Review*, 79(5):845–856, 1950.

[50] Lev Davidovich Landau and VL Ginzburg. On the theory of superconductivity. *Zh. Eksp. Teor. Fiz.*, 20:1064, 1950.

[51] PG De Gennes. *Superconductivity of Metals and Alloys (Advanced Book Classics)*. Addison-Wesley Publ. Company Inc, 1999.

[52] A. A. Golubov, M. Yu. Kupriyanov, and E. Il'ichev. The current-phase relation in josephson junctions. *Rev. Mod. Phys.*, 76:411–469, Apr 2004.

[53] KK Likharev. Superconducting weak links. *Reviews of Modern Physics*, 51(1):101, 1979.

[54] L Ponta, A Carbone, M Gilli, and P Mazzetti. Resistive transition in granular disordered high Tc superconductors: A numerical study. *Physical Review B*, 79(13):134513, April 2009.

[55] SK Tolpygo, S Shokhor, B Nadgorny, J-Y Lin, M Gurvitch, A Bourdillon, SY Hou, and Julia M Phillips. High quality yba2cu3o7 josephson junctions made by direct electron beam writing. *Applied physics letters*, 63(12):1696–1698, 1993.

[56] Shane A Cybart, SM Wu, SM Anton, I Siddiqi, John Clarke, and RC Dynes. Series array of incommensurate superconducting quantum interference devices from yba2cu3o7- δ ion damage josephson junctions. *Applied Physics Letters*, 93(18):182502, 2008.

[57] C Lebeau, J Rosenblatt, A Raboutou, and P Peyral. Current-voltage hyperscaling in arrays of josephson junctions. *EPL (Europhysics Letters)*, 1(6):313, 1986.

[58] MV Simkin and JM Kosterlitz. Finite size and current effects on iv characteristics of josephson junction arrays. *Physical Review B*, 55(17):11646, 1997.

[59] P Martinoli and C Leemann. Two dimensional Josephson junction arrays. *Journal of Low Temperature Physics*, pages 699–731, 2000.

[60] E Granato and D Domínguez. Current-voltage characteristics of diluted Josephson-junction arrays: Scaling behavior at current and percolation threshold. *Physical Review B*, 56(22):671–676, 1997.

[61] A-L Eichenberger, J Affolter, M Willemin, M Mombelli, H Beck, P Martinoli, and SE Korshunov. Dynamic measurement of percolative critical exponents in disordered josephson junction arrays. *Physical review letters*, 77(18):3905, 1996.

[62] M Octavio, A Octavio, J Aponte, R Medina, and CJ Lobb. Nonuniversal critical behavior in the critical current of superconducting composites. *Physical Review B*, 37(16):9292, 1988.

[63] Kh Lee and D Stroud. Critical currents of composite superconductors: Model calculations. *Physical review. B, Condensed matter*, 45(5):2417–2424, February 1992.

[64] Alexey Bezryadin. Quantum suppression of superconductivity in nanowires. *Journal of Physics: Condensed Matter*, 20(4):043202, 2008.

[65] VL Berezinskii. Destruction of long-range order in one-dimensional and two-dimensional systems having a continuous symmetry group i. classical systems. *Sov. Phys. JETP*, 32(3):493–500, 1971.

[66] John Michael Kosterlitz and David James Thouless. Ordering, metastability and phase transitions in two-dimensional systems. *Journal of Physics C: Solid State Physics*, 6(7):1181, 1973.

[67] MR Beasley, JE Mooij, and TP Orlando. Possibility of vortex-antivortex pair dissociation in two-dimensional superconductors. *Physical Review Letters*, 42(17):1165, 1979.

[68] A Bunde and Antonio Coniglio. Transport in a two-component randomly composite material: scaling theory and computer simulations of termite diffusion near the superconducting limit. *Journal of Physics A: Mathematical and General*, 137, 1985.

[69] JP Straley. Non-universal threshold behaviour of random resistor networks with anomalous distributions of conductances. *Journal of Physics C: Solid State Physics*, 2343, 1982.

[70] Scott Kirkpatrick. Percolation and conduction. *Reviews of modern physics*, 45, 1973.

[71] JM Luck. Conductivity of random resistor networks: An investigation of the accuracy of the effective-medium approximation. *Physical Review B*, 43(5), 1991.

[72] AB Harris, Yigal Meir, and Amnon Aharony. Resistance distributions of the random resistor network near the percolation threshold. *Physical Review B*, 41(7):4610–4618, 1990.

[73] DC Wright, DJ Bergman, and Y Kantor. Resistance fluctuations in random resistor networks above and below the percolation threshold. *Physical Review B*, 33(1), 1986.

[74] Robert Haslinger and Robert Joynt. Theory of percolative conduction in polycrystalline high-temperature superconductors. *Physical Review B*, 61(6):4206–4214, February 2000.

[75] M Prester. Dynamical exponents for the current-induced percolation transition in high-Tc superconductors. *Physical review. B, Condensed matter*, 54(1):606–618, July 1996.

[76] Hisao Nakanishi. Critical behaviour of ab percolation in two dimensions. *Journal of Physics A: Mathematical and General*, 20(17):6075, 1987.

[77] John C Wierman. Ab percolation: a brief survey. *Banach Center Publications*, 25(1):241–251, 1989.

[78] A Puri and T Odagaki. Hopping conduction in insulator-conductor-superconductor mixtures. *Journal of Physics A: Mathematical and General*, 22(17):3681, 1989.

[79] Massimiliano Giona and Alessandra Adrover. Multicomponent percolation: Probabilistic properties and application to nonisothermal reactions in granular materials. *Physical Review E*, 49(6):5287, 1994.

[80] Alexei S Ioselevich. Multicomponent percolation criterion and its application to hopping in disordered conductors. *Physical review letters*, 74(8):1411, 1995.

[81] Paul M Kogut and Joseph P Straley. The bicritical macroscopic conductivity exponent in dimensionalities two, three and infinity. *Journal of Physics C: Solid State Physics*, 12(1):1, 1979.

[82] Sergey V Barabash, David J Bergman, and D Stroud. Magnetoresistance of three-constituent composites: Percolation near a critical line. *Physical Review B*, 64(17):174419, 2001.

[83] Ronen Magier and David J. Bergman. Strong-field magnetotransport in a normal conductor/perfect conductor/insulator disordered composite material: Simulations of a discrete model. *Phys. Rev. B*, 83:174445, May 2011.

[84] W Xia and Pl Leath. Defects, vortices, and critical current in Josephson-junction arrays. *Physical review letters*, 63(13):1428–1431, September 1989.

[85] XC Zeng, D Stroud, and JS Chung. Monte Carlo studies of percolation in a superconducting array. *Physical Review B*, 43(4), 1991.

[86] Young-Je Yun, In-Cheol Baek, and Mu-Yong Choi. Phase transition and critical dynamics in site-diluted josephson-junction arrays. *Physical review letters*, 97(21):215701, 2006.

[87] Linda Ponta, Anna Carbone, Marco Gilli, and Piero Mazzetti. Resistively and capacitively shunted josephson junctions model for unconventional superconductors. In *Electrical and Computer Engineering (CCECE), 2011 24th Canadian Conference on*, pages 000644–000647. IEEE, 2011.

[88] Linda Ponta, Valentina Andreoli, and Anna Carbone. Superconducting-insulator transition in disordered josephson junctions networks. *arXiv preprint arXiv:1304.1911*, 2013.

[89] RM Bradley, D Kung, PN Strenski, and S Doniach. Finite-voltage behavior of highly-disordered granular superconductors. *Physica B: Condensed Matter*, 152:282–287, 1988.

[90] R Riedinger. Critical current simulation in granular superconductors above the percolation threshold. *Journal de Physique III*, 1992.

[91] Pedro A Pury and Manuel O Caceres. Tunneling percolation model for granular metal films. *Physical Review B*, 55(6):3841, 1997.

[92] Yakov Strelniker, Aviad Frydman, and Shlomo Havlin. Percolation model for the superconductor-insulator transition in granular films. *Physical Review B*, 76(22):224528, December 2007.

[93] JJ Brey, J Gmez Ordez, and A Santos. Diffusion in a 2d bond percolation model. a monte carlo simulation. *Physics Letters A*, 127(1):5 – 8, 1988.

[94] GA Van Velzen and MH Ernst. Bond percolation in two and three dimensions: Numerical evaluation of time-dependent transport properties. *Journal of statistical physics*, 48(3-4):677–708, 1987.

[95] Jan Tobochnik, David Laing, and Gary Wilson. Random-walk calculation of conductivity in continuum percolation. *Physical Review A*, 41(6):3052, 1990.

[96] B Derrida, JG Zabolitzky, J Vannimenus, and D Stauffer. A transfer matrix program to calculate the conductivity of random resistor networks. *Journal of statistical physics*, 36(1-2):31–42, 1984.

[97] B Derrida and J_ Vannimenus. A transfer-matrix approach to random resistor networks. *Journal of Physics A: Mathematical and General*, 15(10):L557, 1982.

[98] JG Zabolitzky. Monte carlo evidence against the alexander-orbach conjecture for percolation conductivity. *Physical Review B*, 30:4077–4079, 1984.

[99] J-M Normand, HJ Herrmann, and M Hajjar. Precise calculation of the dynamical exponent of two-dimensional percolation. *Journal of Statistical Physics*, 52(1-2):441–446, 1988.

[100] CJ Lobb and DJ Frank. Percolative conduction and the alexander-orbach conjecture in two dimensions. *Physical Review B*, 30(7):4090, 1984.

[101] Stéphane Tyč and BI Halperin. Random resistor network with an exponentially wide distribution of bond conductances. *Physical Review B*, 39(1):877, 1989.

[102] Alan R Kerstein. Equivalence of the void percolation problem for overlapping spheres and a network problem. *Journal of Physics A: Mathematical and General*, 16(13):3071, 1983.

[103] Adam M Becker and Robert M Ziff. Percolation thresholds on 2d voronoi networks and delaunay triangulations. *arXiv preprint arXiv:0906.4360*, 2009.

[104] Boris Delaunay. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934.

[105] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition, 2008.

[106] PM Kogut and JP Straley. Distribution-induced non-universality of the percolation conductivity exponents. *Journal of Physics C: Solid State Physics*, 2151, 1979.

[107] M Octavio and CJ Lobb. Nonuniversality in two-dimensional percolating systems with a broad distribution of bond conductances. *Physical Review B*, 43(10):8233, 1991.

[108] XC Zeng, DJ Bergman, and D Stroud. Numerical study of transport properties in continuum percolation. *Journal of Physics A: Mathematical and General*, 21(19):L949, 1988.

[109] MD Rintoul and S Torquato. Algorithm to compute void statistics for random arrays of disks. *Physical Review E*, 52(3), 1995.

[110] JP Clerc, VA Podolskiy, and AK Sarychev. Precise determination of the conductivity exponent of 3d percolation using exact numerical renormalization. *The European Physical Journal B-Condensed Matter and Complex Systems*, 15(3):507–516, 2000.

[111] Shigeki Matsutani, Yoshiyuki Shimosako, and Yunhong Wang. Numerical computations of conductivity in continuum percolation for overlapping spheroids. *International Journal of Modern Physics C*, 21(06):709–729, 2010.

[112] Ying Chen and Christopher A Schuh. Effective transport properties of random composites: Continuum calculations versus mapping to a network. *Physical Review E*, 80(4):040103, 2009.

[113] J Quintanilla and S Torquato. Clustering properties of d-dimensional overlapping spheres. *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, 54(5):5331–5339, November 1996.

[114] Andrew L Dulmage and Nathan S Mendelsohn. Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10(4):516–534, 1958.

[115] Alex Pothen and Chin-Ju Fan. Computing the block triangular form of a sparse matrix. *ACM Transactions on Mathematical Software (TOMS)*, 16(4):303–324, 1990.